University of Latvia

Marats Golovkins

# Quantum Automata and Quantum Computing

Doctoral Thesis

Riga 2002

University of Latvia
Institute of Mathematics and Computer Science

Marats Golovkins

# Quantum Automata and Quantum Computing

Doctoral Thesis

Area: Computer Science
Sub-Area: Mathematical Foundations of Computer Science

Scientific Advisor:
Dr. habil. math., Prof.
**Rūsiņš Freivalds**

Riga 2002

# Abstract

Quantum finite automata were introduced by C. Moore and J. P. Crutchfield in [MC 97] and by A. Kondacs and J. Watrous in [KW 97]. This notion is not a generalization of the deterministic finite automata, but rather a generalization of deterministic reversible (permutation) automata. In [AF 98] A. Ambainis and R. Freivalds raised the question what kind of probabilistic automata can be viewed as a special case of quantum finite automata. To answer that question and study relationship between quantum finite automata and probabilistic finite automata, we introduce a notion of probabilistic reversible automata (PRA, or doubly stochastic automata). We give the necessary condition for a language to be recognized by PRA. We find that there is a strong relationship between different possible models of PRA and corresponding models of quantum finite automata.

In these thesis we regard quantum automata, probabilistic reversible and deterministic reversible automata as reversible automata. At least two non-equivalent definitions of language recognition are used for one-way reversible finite automata in various papers. Both of these definitions have their own advantages and disadvantages, summarized in our classification of one-way reversible finite automata.

Further, we introduce a notion of quantum finite multitape automata and prove that there is a language recognized by a quantum finite multitape automaton but not by deterministic finite multitape automata. Additionally we discover unexpected probabilistic automata recognizing complicated languages.

Finally, we revise the concept of quantum pushdown automata (QPA), first introduced by C. Moore and J. P. Crutchfield in [MC 97]. We give the definition of QPA in a non-equivalent way, including unitarity criteria, by using the definition of quantum finite automata of [KW 97]. It is established that the unitarity criteria of QPA are not equivalent to the corresponding unitarity criteria of quantum Turing machines [BV 97]. We show that QPA can recognize every regular language. We also present some simple non-context-free languages recognized by QPA (hence not recognizable by deterministic and nondeterministic pushdown automata).

# Anotācija

Kvantu galīgā automāta definīciju ieviesa K. Mūrs un Dž. P. Kračfīlds rakstā [MC 97], kā arī A. Kondačs un Dž. Vatro rakstā [KW 97]. Šis jēdziens nav galīgo determinēto automātu vispārinājums, bet gan drīzāk determinēto apgriežamo (permutāciju) automātu vispārinājums. Rakstā [AF 98] A. Ambainis un R. Freivalds uzdeva jautājumu, kāda tipa varbūtiskie automāti var tikt aplūkoti kā kvantu galīgo automātu speciālgadījums. Lai atbildētu uz šo jautājumu un pētītu saistību starp kvantu galīgajiem automātiem un varbūtiskajiem galīgajiem automātiem, ieviests varbūtiskā apgriežamā automāta (dubultstohastiskā automāta) jēdziens. Dots nepieciešamais nosacījums, lai valodu varētu atpazīt ar varbūtisko apgriežamo automātu palīdzību. Atrasts, ka pastāv ciešs sakars starp dažādiem iespējamiem varbūtisko apgriežamo automātu modeļiem un atbilstošajiem kvantu galīgo automātu modeļiem.

Šajā disertācijā par apgriežamajiem automātiem uzskatīti kvantu automāti, varbūtiskie apgriežamie un determinētie apgriežamie automāti. Valodu pazīšanai ar vienvirziena apgriežamajiem galīgajiem automātiem dažādos rakstos tiek izmantotas vismaz divas neekvivalentas definīcijas. Abām šīm definīcijām ir savas priekšrocības un savi trūkumi, kas apkopoti vienvirziena apgriežamo galīgo automātu klasifikācijā.

Tālāk, ieviests kvantu galīgā daudzlenšu automāta jēdziens un pierādīts, ka eksistē valoda, ko var pazīt ar kvantu galīgo daudzlenšu automātu, bet ne ar determinētajiem galīgajiem daudzlenšu automātiem. Papildus atklāti negaidīti varbūtiskie automāti, kas pazīst sarežģītas valodas.

Visbeidzot, pārskatīts kvantu automāta ar magazīnas tipa atmiņu jēdziens, ko pirmoreiz ieviesa K. Mūrs un Dž. P. Kračfīlds rakstā [MC 97]. Dota kvantu automāta ar magazīnas tipa atmiņu definīcija, kas nav ekvivalenta ar iepriekšminēto, izmantojot kvantu galīgā automāta definīciju no raksta [KW 97]. Noteikts, ka kvantu automāta ar magazīnas tipa atmiņu unitaritātes kritērijs nav ekvivalents ar atbilstošo kvantu Tjūringa mašīnas unitaritātes kritēriju rakstā [BV 97]. Pierādīts, ka kvantu automāti ar magazīnas tipa atmiņu pazīst jebkuru regulāru valodu. Tāpat uzrādītas dažas vienkāršas valodas, kuras nav bezkonteksta (līdz ar to tās nepazīst determinētie un nedeterminētie automāti ar magazīnas tipa atmiņu) un kuras pazīst kvantu automāti ar magazīnas tipa atmiņu.

# Аннотация

Определение квантовых конечных автоматов ввели К. Мур и Дж. П. Крачфилд в статье [MC 97], а также А. Кондач и Дж. Ватро в статье [KW 97]. Это понятие не является обобщением детерминированных конечных автоматов, а скорее это обобщение детерминированных обратимых (пермутационных) автоматов. В статье [AF 98] А. Амбаинис и Р. Фрейвалдс подняли вопрос, какие вероятностные автоматы могут считаться частным случаем квантовых конечных автоматов. Чтобы ответить на этот вопрос и изучить связь между квантовыми конечными автоматами и вероятностными конечными автоматами, введено понятие вероятностного обратимого (дважды стохастического) автомата. Представлено необходимое условие, чтобы язык был бы распознаваем вероятностными обратимыми автоматами. Найдено, что существует тесная связь между различными возможными моделями вероятностных обратимых автоматов и соответствующими моделями квантовых конечных автоматов.

В данной диссертации обратимыми автоматами считаются квантовые автоматы, вероятностные обратимые и детерминированные обратимые автоматы. В различных статьях для распознавания языков односторонними обратимыми конечными автоматами используются как минимум два неравносильных определения. Оба определения имеют свои преимущества и недостатки, что обобщено в классификации односторонних обратимых конечных автоматов.

Далее, введено понятие квантового многоленточного конечного автомата и доказано, что существует язык, который возможно распознать квантовым конечным многоленточным автоматом, но не детерминированными конечными автоматами. Дополнительно найдены неожиданные вероятностные автоматы, распознающие сложные языки.

Наконец, пересмотрено понятие квантового автомата с магазинной памятью, впервые введённым К. Муром и Дж. П. Крачфилдом в статье [MC 97]. Дано определение квантового автомата с магазинной памятью, которое неэквивалентно с вышеуказанным, используя определение квантового конечного автомата от [KW 97]. Установлено, что критерий унитарности квантового автомата с магазинной памятью неравносилен соответствующему критерию унитарности квантовой машины Тюринга [BV 97]. Доказано, что квантовые автоматы с магазинной памятью распознают любой регулярный язык. Также представлены некоторые простые языки, которые не являются контексто-свободными (следовательно, не распознаваемы детерминированными и недетерминированными автоматами с магазинной памятью) и распознаваемы квантовыми автоматами с магазинной памятью.

# Preface

This thesis assembles the research performed by the author and reflected in the following publications:

1. M. Golovkins, M. Kravtsev. Probabilistic Reversible Automata and Quantum Automata. *COCOON 2002, Lecture Notes in Computer Science*, Vol. 2387, pp. 574-583, 2002.

2. M. Golovkins. Quantum Pushdown Automata. *SOFSEM 2000, Lecture Notes in Computer Science*, Vol. 1963, pp. 336-346, 2000.

3. A. Ambainis, R. Bonner, R. Freivalds, M. Golovkins, M. Karpinski. Quantum Finite Multitape Automata. *SOFSEM 1999, Lecture Notes in Computer Science*, Vol. 1725, pp. 340-348, 1999.

4. M. Golovkins, M. Kravtsev. Probabilistic Reversibility and Its Relation to Quantum Automata. *Quantum Computation and Learning. 3rd International Workshop. Proceedings*, pp. 1-22, Riga, 2002.

5. M. Golovkins. On Quantum Pushdown Automata. *Quantum Computation and Learning. 2nd International Workshop. Proceedings*, pp. 41-51, Sundbyholms Slott, Sweden, 2000.

6. A. Ambainis, R. Bonner, R. Freivalds, M. Golovkins, M. Karpinski. Quantum vs Probabilistic Finite Multitape Automata. *Quantum Computation and Learning. 1st International Workshop. Proceedings*, pp. 36-43, Riga, 1999.

7. R. Bonner, R. Freivalds, M. Golovkins. Projections of Multitape Languages Recognized by Quantum and Probabilistic Finite Automata. *Quantum Computation and Learning. 1st International Workshop. Proceedings*, pp. 84-92, Riga, 1999.

8. M. Golovkins. An Introduction to Quantum Pushdown Automata. *Quantum Computation and Learning. 1st International Workshop. Proceedings*, pp. 44-52, Riga, 1999.

The results of the thesis were presented at the following international conferences and workshops:

1. Computing and Combinatorics. 8th Annual International Conference, COCOON 2002, Singapore, August 15-17. Presentation "Probabilistic Reversible Automata and Quantum Automata".

2. Euroworkshop on Quantum Computers: Mesoscopic Implementation; Perspectives and Open Problems. Villa Gualino, Torino, Italy, June 10-21, 2002. Presentation "Quantum Automata and Probabilistic Reversible Automata".

3. Quantum Computation and Learning. 3rd International Workshop. Riga, Latvia, May 25-26, 2002. Presentation "Quantum Automata and Probabilistic Reversible Automata".

4. Euroworkshop on Quantum Computer Theory: In Search of Viable Optimal Design. Villa Gualino, Torino, Italy, June 18-30, 2001. Presentation "Quantum Finite Automata with Pure States". Co-presented by R. Freivalds, A. Ķikusts.

5. SOFSEM 2000: Theory and Practice of Informatics. 27th Conference on Current Trends in Theory and Practice of Informatics. Milovy, Czech Republic, November 25 - December 2. Presentation "Quantum Pushdown Automata".

6. Quantum Computation and Learning. 2nd International Workshop. Sundbyholms Slott, Sweden, May 27-29, 2000. Presentation "On Quantum Pushdown Automata".

7. International Workshop "Quantum Days in Växjö". University of Växjö, Växjö, Sweden, December 9-10, 1999. Presentation "On Quantum Finite Multitape Automata".

8. SOFSEM'99: Theory and Practice of Informatics. 26th Conference on Current Trends in Theory and Practice of Informatics. Milovy, Czech Republic, November 27 - December 4. Presentation "Quantum Finite Multitape Automata".

9. Quantum Computation and Learning. 1st International Workshop. Riga, Latvia, September 11-13, 1999. Presentations "Quantum vs Probabilistic Finite Multitape Automata" and "An Introduction to Quantum Pushdown Automata".

I thank my co-authors Andris Ambainis, Richard Bonner, Rūsiņš Freivalds, Marek Karpinski, Maksim Kravtsev, who significantly contributed to this research. I thank Arnolds Ķikusts for useful discussions.

Especially I thank my supervisor Prof. Rūsiņš Freivalds, whose encouragement meant great deal for me in pursuing the subject.

# Contents

# Chapter 1

# Introduction

Nobel prize winner physicist Richard Feynman asked in [Fe 82] what effects may have the principles of quantum mechanics, especially, the principle of superposition on computation. He gave arguments that it may require exponential time to simulate quantum mechanical processes on classical computers. This served as a basis to the opinion that quantum computers may have advantages over classical ones. It was in 1985, when D. Deutsch introduced the notion of quantum Turing machine [De 85] and proved that quantum Turing machines compute exactly the same recursive functions as classical deterministic Turing machines do. However, P. Shor discovered that by use of quantum algorithms it is possible to factorize large integers and compute discrete logarithms in a polynomial time [Sh 94], what resulted into additional interest in quantum computing and attempts to build quantum computers. First steps have been made on this direction, and first quantum computers which memory is limited by a few quantum bits have been constructed. To make quantum computers with larger memory feasible, one of the problems is to minimize error possibilities in quantum bits. Quantum error correction methods are developed which would enable quantum computers with larger quantum memory.

Quantum computation and information processing as well as main open issues are analyzed in [Gr 99].

Quantum mechanics differs from the classical physics substantially. It is enough to mention *Heisenberg's uncertainty principle*, which states that it is impossible to get information about different parameters of quantum particle simultaneously precisely. Another well known distinction is the impossibility to observe quantum object without changing it.

Fundamental concept of quantum information theory is *quantum bit.* Classical information theory is based on classical bit, which has two states 0 and 1. The next step is *probabilistic bit*, which can be 0 with probability $\alpha$

and 1 with probability $\beta$, where $\alpha + \beta = 1$. Quantum bit or *qbit* is similar to probabilistic bit with the difference that $\alpha$ and $\beta$ are complex numbers with the property $|\alpha|^2 + |\beta|^2 = 1$. It is common to denote qbit as $\alpha|0\rangle + \beta|1\rangle$. As a result of *measurement*, we get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.

Every computation done on qbits is accomplished by means of unitary operators. Informally, every unitary operator can be interpreted as a rotation in complex space. Therefore one of the basic properties of unitary operators is that every quantum computing process not disturbed by measurements is reversible. Unitarity is rather hard requirement which complicates programming of quantum devices. Still it is possible to simulate every classical Turing machine by quantum Turing machine with only polynomial slowdown. The following features of quantum computers are most important:

1. Information is represented by qbits.
2. Any step of computation can be represented as a unitary operation, therefore computation is reversible.
3. Quantum information cannot be copied.
4. Quantum parallelism; quantum computer can compute several paths simultaneously, however as a result of measurement it is possible to get the results of only one computation path.

As in the classical theory of computation, we may consider other models of computation, such as quantum finite automata, pushdown automata, etc. Opposite to Turing machines, these models represent the cases when specific restrictions are inevitable regarding space consumption or time usage. In particular, one-way quantum finite automata (1-QFA) represent one of the most restricted models, where computation is performed with finite memory and in real time, i.e., the number of computation steps does not exceed the length of input.

In quantum computation, it is possible to distinguish between pure state model, where an automaton is in a single quantum superposition of configurations after each computation step, and mixed state model, where the automaton with certain probabilities is in one of several possible quantum superpositions.

In these thesis, we study quantum automata in terms of formal languages they can recognize. Similarly as in probabilistic computation it is possible to consider language recognition with bounded error or unbounded error, however we restrict ourselves to bounded error language recognition only.

We shall understand under reversible automata quantum automata, probabilistic reversible automata, as well as deterministic reversible (permutation) automata.

Several definitions are used in various papers for language recognition by reversible automata.

Generally, an automaton halts computation (enters a halting configuration) and then gives an answer whether a word is in the language (a halting configuration may be either accepting or rejecting). We refer to this type of language recognition as "decide and halt" and to the respective automata as "decide and halt" automata (DH-automata). In case of quantum automata, to know whether to continue computation or not, we must constantly check whether a quantum automaton has entered a halting configuration, hence respective measurements are performed after each step[1]. As soon as the result of measurement means obtaining a superposition of halting configurations, we measure whether a configuration is accepting or rejecting one[2]. If no other type of measurements is allowed after each step, we have pure state model, otherwise we obtain mixed state model.

For real time automata, it is possible to have other definition where there is no notion of the halting configuration; an automaton halts computation as soon as the number of computation steps equals length of the input instead, and then gives an answer whether a word is in the language (any configuration is either accepting or rejecting). This definition is widely used with classical one-way deterministic and probabilistic finite automata, hence we refer to this type of language recognition as classical and to the respective automata as classical automata (C-automata). In case of quantum automata, no measurements are really necessary after each step; at the end of computation a measurement whether a configuration is accepting or rejecting is performed. If some other type of measurements is still allowed after each step, we obtain mixed state model, otherwise we have pure state model.

For classical one-way deterministic and probabilistic finite automata the both definitions are equivalent. However, if one-way reversible finite automata are considered, the "decide and halt" definition is more powerful, yet with less pleasant theoretical properties.

It is still possible to generalize the classical definition even if computation is not real time, since we can define a function, where the argument is length of input and the output is allowed number of computation steps. Such function could be applied to the "decide and halt" definition as well. However, all

---

[1] It is possible to perform measurements so that quantum superposition among non-halting states is not disturbed.

[2] Actually, these two types of measurements may be combined to one single type of measurement.

that is beyond the limits of this thesis. Eventually, we consider the classical definition as well as the "decide and halt" definition in the case of one-way reversible finite automata and use the "decide and halt" definition for other types of automata.

Quantum Turing machine was introduced by D. Deutsch in [De 85]. Classical 1-QFA with pure states were introduced by C. Moore, J. P. Crutchfield in [MC 97]. Subsequently, A. Kondacs, and J. Watrous introduced "decide and halt" 1-QFA with pure states in [KW 97]. Classical 1-QFA with pure states and "decide and halt" 1-QFA with pure states are commonly referred in literature as *measure-once* QFA (MO-QFA) and *measure-many* QFA (MM-QFA), respectively. Since then 1-QFA with pure states have been studied a lot, measure-once finite automata are considered in [AF 98, BP 99, AG 00, BB 01], whereas measure-many automata in [AF 98, ANTV 98, ABFK 99, BP 99, AKV 00, AK 01, AIR 02, M 02]. In particular, Kondacs and Watrous showed in [KW 97], that MM-QFA can recognize only a proper subset of regular languages. However, in [AF 98] Ambainis and Freivalds showed that for some languages QFA may be exponentially more concise. Still, Ambainis, Nayak et al. pointed out in [ANTV 98] that for some other languages, QFA may require exponentially larger number of states than deterministic automata do. In [BP 99], Brodsky and Pippenger noted that MO-QFA recognize the same language class as permutation automata ([T 68]). Ambainis, Ķikusts and Valdats determined in [AKV 00] that the class of languages recognized by MM-QFA is not closed under boolean operations, as well as significantly improved the necessary condition of a language to be recognized by MM-QFA, proposed by [BP 99]. In [AK 01], Ambainis and Ķikusts gave the final answer what is the largest probability for a language to be recognized by MM-QFA, if it is not recognizable by deterministic reversible ("decide and halt") finite automata.

"Decide and halt" 1-QFA with mixed states were introduced by A. Nayak in [N 99] as *enhanced* quantum finite automata. He showed that the similar weaknesses apply to this more general model as shown for MM-QFA. Still the language class and other properties of the model are not much explored. One chapter in this thesis is devoted to probabilistic reversible automata [GK 02], which with some additional restrictions are one of the marginal special cases of Nayak's model (the other marginal special cases are MO-QFA and MM-QFA).

2-way quantum finite automata are considered in [KW 97, AI 99, AW 99, Du 01]. Kondacs and Watrous brought in 2-way quantum finite automata and proved that a non-regular language can be recognized by this model in a quadratic time. (Probabilistic 2-way automata can do that in exponential time only, see [Fr 81, DSt 89].) In [AI 99], Amano and Iwama showed that emptiness problem is not decidable for 2-way (even 1.5-way) quantum finite automata. In [Du 01] it is actually determined that even 1.5-way QFA can recognize non-regular languages, which is not possible by deterministic and probabilistic counterparts of this model.

The paper [ABFGK 99] discusses the properties of quantum finite multitape automata. Quantum 1-counter automata are explored in [K 99, YKTI 00, BFK 01, YKI 01, YKI 02], whereas quantum pushdown automata are considered in [MC 97, Go 00, NIHK 01]. In [Go 00], it was proved that quantum pushdown automata can recognize every regular language using pushdown store as a recording device to maintain reversibility. Quantum finite state transducers are introduced in [FW 01]. It was shown that quantum finite state transducers can compute relations, which are not computable by deterministic or probabilistic finite state transducers. In [PC 01], it is actually noted that quantum finite state transducers may be used to recognize every regular language, the idea is similar as in [Go 00]. In [Sc 01], quantum real time Turing machines are considered.

In Chapter 2 of the thesis we state common notations and definitions, used in the rest of the chapters. In Section 2.1 we recall several notions of linear algebra, whereas in Section 2.2 discuss notions applicable to arbitrary automata model.

In Chapter 3 we introduce and research probabilistic reversible automata and outline the fundamental relations between this model and other models of one-way reversible finite automata. In Section 3.1, we discuss properties of probabilistic reversible C-automata (C-PRA). We prove that C-PRA recognize the class of languages $a_1^*a_2^* \dots a_n^*$ with probability $1-\varepsilon$. This class can be recognized by MM-QFA, with worse acceptance probabilities, however [ABFK 99]. This also implies that Nayak's enhanced QFA recognize this class of languages with probability $1-\varepsilon$. Further, we show general class of regular languages, not recognizable by C-PRA. In particular, such languages as (a,b)*a and a(a,b)* are in this class. This class has strong similarities with the class of languages, not recognizable by MM-QFA [AKV 00]. We also show that the class of languages recognized by C-PRA is closed under boolean operations, inverse homomorphisms and word quotient, but is not closed under homomorphisms. In Section 3.2 we prove that C-PRA automata without end-markers recognize the same class of languages as C-PRA automata with both end-markers. In Section 3.3 we propose a classification of one-way reversible finite automata (deterministic, probabilistic and quantum).

In Chapter 4 we deal with quantum finite multitape automata (QFMA). In Section 4.1 we define QFMA. In Section 4.2 we define language recognition and explore languages recognized by QFMA in comparison with deterministic and probabilistic finite multitape automata.

In Chapter 5 we explore the basic properties of quantum pushdown automata (QPA). In Section 5.1 we advocate the notion of QPA and give a formal definition of this model. In Section 5.2 we explore languages recognized by QPA. We prove that QPA can recognize every regular language.

Chapter 6 is the conclusion, which summarizes the work and states main open problems.

# Chapter 2

# Preliminaries

The following notations will be used further in the thesis: $z^*$ is the complex conjugate of a complex number $z$, $U^*$ is the Hermitian conjugate of a matrix $U$, $S^T$ is the transpose of a matrix $S$, $I$ is the identity matrix, $Q$ is the set of states of an automaton, $\epsilon$ is an empty word, $\Sigma^*$ is the set of all finite words over alphabet $\Sigma$, a language $L$ is a subset of $\Sigma^*$, $\overline{L}$ is a complement of the language $L$. Given a word $\omega \in \Sigma^*$, $|\omega|$ is the number of symbols in $\omega$ and $[\omega]_i$ is the i-th symbol of $\omega$, starting counting with 1 from the left. A reverse of a word $\omega = a_1 a_2 \dots a_{n-1} a_n$, where $a_i \in \Sigma$, is the word $\omega^R = a_n a_{n-1} \dots a_2 a_1$. A reverse of a language $L \subset \Sigma^*$ is the language $L^R = \{x \in \Sigma^* \mid x^R \in L\}$.

## 2.1 Unitary and Stochastic Operations

In this section, we recall well known definitions and theorems from linear algebra and Markov chains theory. For the sake of completeness, some of the theorems are supplied with elementary proofs.

**Definition 2.1.** *We say that a* $(kn \times kn)$ *matrix* $C$ *is a direct product of* $(k \times k)$ *matrix* $A$ *and* $(n \times n)$ *matrix* $B$, $C = A \bigotimes B$, *if* $c_{(i-1)n+m,(j-1)n+l} = a_{i,j} b_{m,l}$.

### Unitary Matrices

**Definition 2.2.** *A complex matrix* $U$ *is called* unitary, *if* $UU^* = U^*U = I$.

**Lemma 2.3.** *For arbitrary integer $n > 0$, the matrix*

$$\begin{pmatrix} \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}} & \dots & \frac{1}{\sqrt{n}} \\ \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}}e^{\frac{2\pi i}{n}} & \frac{1}{\sqrt{n}}e^{\frac{4\pi i}{n}} & \dots & \frac{1}{\sqrt{n}}e^{\frac{2\pi(n-1)i}{n}} \\ \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}}e^{\frac{4\pi i}{n}} & \frac{1}{\sqrt{n}}e^{\frac{8\pi i}{n}} & \dots & \frac{1}{\sqrt{n}}e^{\frac{4\pi(n-1)i}{n}} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{\sqrt{n}} & \frac{1}{\sqrt{n}}e^{\frac{2\pi(n-1)i}{n}} & \frac{1}{\sqrt{n}}e^{\frac{4\pi(n-1)i}{n}} & \dots & \frac{1}{\sqrt{n}}e^{\frac{2\pi(n-1)^2 i}{n}} \end{pmatrix}$$

*is unitary.*

By this lemma, quantum automata are able to make equiprobable choices among a finite number of possibilities.

**Lemma 2.4.** *If matrices $A$ and $B$ are unitary, then their direct product is a unitary matrix.*

If $U$ is a finite matrix, then $UU^* = I$ iff $U^*U = I$. However this is not true for infinite matrices:

**Example 2.5.**

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \dots \\ \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Here $U^*U = I$ but $UU^* \neq I$.

**Lemma 2.6.** *If infinite matrices $A, B, C$ have finite number of nonzero elements in each row and column, then their multiplication is associative: $(AB)C = A(BC)$.*

*Proof.* The element of matrix $(AB)C$ in $i$-th row and $j$-th column is $k_{ij} = \sum\limits_{s=1}^{\infty}\sum\limits_{r=1}^{\infty} a_{ir}b_{rs}c_{sj}$. The element of matrix $A(BC)$ in the same row and column is $l_{ij} = \sum\limits_{r=1}^{\infty}\sum\limits_{s=1}^{\infty} a_{ir}b_{rs}c_{sj}$. As in the each row and column of matrices $A, B, C$ there is a finite number of nonzero elements, it is also finite in the given series. Therefore the elements of the series can be rearranged, and $k_{ij} = l_{ij}$. □

As noted in the next section infinite unitary matrices with finite number of nonzero elements in each row and column describe the work of quantum automata. Further lemmas state some properties of such matrices.

**Lemma 2.7.** *If $U^*U = I$, then the norm of any row in the matrix $U$ does not exceed* 1.

*Proof.* Let us consider the matrix $S = UU^*$. The element of this matrix $s_{ij} = \langle r_j | r_i \rangle$, where $r_i$ is $i$-th row of the matrix $U$. Let us consider the matrix $T = S^2$. The diagonal element of this matrix is

$$t_{ii} = \sum_{k=1}^{\infty} s_{ik} s_{ki} = \sum_{k=1}^{\infty} \langle r_k | r_i \rangle \langle r_i | r_k \rangle = \sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2.$$

On the other hand, taking into account Lemma 2.6, we get that

$$T = S^2 = (UU^*)(UU^*) = U(U^*U)U^* = UU^* = S.$$

Therefore $t_{ii} = s_{ii} = \langle r_i | r_i \rangle$. It means that

$$\sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2 = \langle r_i | r_i \rangle. \tag{2.1}$$

This implies that every element of series (2.1) does not exceed $\langle r_i | r_i \rangle$. Hence $|\langle r_i | r_i \rangle|^2 = \langle r_i | r_i \rangle^2 \leq \langle r_i | r_i \rangle$. The last inequality implies that $0 \leq \langle r_i | r_i \rangle \leq 1$. Therefore $|r_i| \leq 1$. □

**Lemma 2.8.** *Let us assume that $U^*U = I$. Then the rows of the matrix $U$ are orthogonal iff every row of the matrix has norm* 0 *or* 1.

*Proof.* Let us assume that the rows of the matrix $U$ are orthogonal. Let us consider equation (2.1) from the proof of Lemma 2.7, i.e., $\sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2 = \langle r_i | r_i \rangle$. As the rows of the matrix $U$ are orthogonal, $\sum_{k=1}^{\infty} |\langle r_k | r_i \rangle|^2 = |\langle r_i | r_i \rangle|^2$. Hence $\langle r_i | r_i \rangle^2 = \langle r_i | r_i \rangle$, i.e., $\langle r_i | r_i \rangle = 0$ or $\langle r_i | r_i \rangle = 1$. Therefore $|r_i| = 0$ or $|r_i| = 1$.

Let as assume that every row of the matrix has norm 0 or 1. Then $\langle r_i | r_i \rangle^2 = \langle r_i | r_i \rangle$ and in compliance with the equation (2.1), $\sum_{k \in \mathbb{N}^+ \setminus \{i\}} |\langle r_k | r_i \rangle|^2 = 0$. This implies that $\forall k \neq i \ |\langle r_k | r_i \rangle| = 0$. Hence the rows of the matrix are orthogonal. □

**Lemma 2.9.** *The matrix $U$ is unitary iff $U^*U = I$ and its rows are normalized.*

*Proof.* Let us assume that the matrix $U$ is unitary. Then in compliance with Definition 2.2, $U^*U = I$ and $UU^* = I$, i.e, the rows of the matrix are orthonormal.

Let us assume that $U^*U = I$ and the rows of the matrix are normalized. Then in compliance with Lemma 2.8 the rows of the matrix are orthogonal. Hence $UU^* = I$ and the matrix is unitary. □

This result is very similar to Lemma 1 of [DSa 96].

## Doubly Stochastic Matrices

**Definition 2.10.** *A real* $(n \times n)$ *matrix* $S$, $s_{i,j} \geq 0$, *is called* stochastic, *if* $\forall j \ \sum\limits_{i=1}^{n} s_{i,j} = 1$.

**Definition 2.11.** *A stochastic* $n \times n$ *matrix* $D$ *is called* doubly stochastic, *if* $\forall i \ \sum\limits_{j=1}^{n} d_{i,j} = 1$.

**Lemma 2.12.** *If matrices $A$ and $B$ are doubly stochastic, then their direct product is a doubly stochastic matrix.*

**Lemma 2.13.** *If $A$ is a doubly stochastic matrix and $X$ - a vector with components* $x_i \geq 0$, *then* $\max(X) \geq \max(AX)$ *and* $\min(X) \leq \min(AX)$.

*Proof.* The idea of the proof due to M. Kravtsev. Let us consider $X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}$ and $A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$, where $A$ is doubly stochastic. Let us suppose that $x_j = \max(X)$. For any $i$, $1 \leq i \leq n$,

$$x_j = a_{i1}x_j + a_{i2}x_j + \ldots + a_{in}x_j \geq a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n.$$

Therefore $x_j$ is greater or equal than any component of $AX$. The second inequality is proved in the same way. □

**Definition 2.14.** *We say that a doubly stochastic matrix $S$ is* unitary stochastic ([MO 79]), *if exists a unitary matrix $U$ such that* $\forall i,j \ |u_{i,j}|^2 = s_{i,j}$.

**Remark 2.15.** Not every doubly stochastic matrix is unitary stochastic. Such matrix is, for example, $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$.

## Markov Chains

We recall several definitions and facts from the theory of finite Markov chains ([KS 76], etc.)

A Markov chain with $n$ states is determined by an $n \times n$ stochastic matrix $A$. If $A_{i,j} = p > 0$, it means that a state $q_i$ is accessible from a state $q_j$ with a positive probability $p$ in one step. Generally speaking, the matrix depends on the numbering of the states; if the states are renumbered, the matrix changes, as its rows and columns also need to be renumbered.

**Definition 2.16.** *A state $q_j$ is accessible from $q_i$ (denoted $q_i \to q_j$) if there is a positive probability to get from $q_i$ to $q_j$ (possibly in several steps).*

**Definition 2.17.** *States $q_i$ and $q_j$ communicate (denoted $q_i \leftrightarrow q_j$) if $q_i \to q_j$ and $q_j \to q_i$.*

**Definition 2.18.** *A state $q$ is called ergodic if $\forall i\ q \to q_i \Rightarrow q_i \to q$. Otherwise the state is called transient.*

**Definition 2.19.** *A Markov chain without transient states is called irreducible if for all $q_i, q_j$ $q_i \leftrightarrow q_j$. Otherwise the chain without transient states is called reducible.*

**Definition 2.20.** *The period of an ergodic state $q_i \in Q$ of a Markov chain with a matrix $A$ is defined as $d(q_i) = \gcd\{n > 0 \mid (A^n)_{i,i} > 0\}$.*

**Definition 2.21.** *An ergodic state $q_i$ is called aperiodic if $d(q_i) = 1$. Otherwise the ergodic state is called periodic.*

**Definition 2.22.** *A Markov chain without transient states is called aperiodic if all its states are aperiodic. Otherwise the chain without transient states is called periodic.*

**Definition 2.23.** *A probability distribution $X$ of a Markov chain with a matrix $A$ is called stationary, if $AX = X$.*

**Definition 2.24.** *A Markov chain is called doubly stochastic, if its transition matrix is a doubly stochastic matrix.*

**Theorem 2.25.** *If a Markov chain with a matrix $A$ is irreducible and aperiodic, then*
*a) it has a unique stationary distribution $Z$;*
*b) $\lim\limits_{n\to\infty} A^n = (Z, \ldots, Z)$;*
*c) $\forall X \lim\limits_{n\to\infty} A^n X = Z$.*

**Corollary 2.26.** *If a doubly stochastic Markov chain with an $m \times m$ matrix $A$ is irreducible and aperiodic,*

*a)* $\lim_{n\to\infty} A^n = \begin{pmatrix} \frac{1}{m} & \dots & \frac{1}{m} \\ \dots & \dots & \dots \\ \frac{1}{m} & \dots & \frac{1}{m} \end{pmatrix}$;

*b)* $\forall X \lim_{n\to\infty} A^n X = \begin{pmatrix} \frac{1}{m} \\ \dots \\ \frac{1}{m} \end{pmatrix}$.

*Proof.* By Theorem 2.25. □

**Lemma 2.27.** *If $M$ is a doubly stochastic Markov chain with a matrix $A$, then $\forall q\ q \to q$.*

*Proof.* The idea of the proof due to M. Kravtsev. Assume existence of $q_0$ such that $q_0$ is not accessible from itself. Let $Q_{q_0} = \{q_i \mid q_0 \to q_i\} = \{q_1, \dots, q_k\}$. $Q_{q_0}$ is not empty set. Consider those rows and columns of $A$, which are indexed by states in $Q_{q_0}$. These rows and columns form a submatrix $A'$. Each column $j$ of $A'$ must include all non-zero elements of the corresponding column of $A$ as those states are accessible from the state $q_j$, hence also from $q_0$ and are in $Q_{q_0}$. Therefore $\forall j$, $1 \le j \le k$, $\sum_{i=1}^{k} a'_{i,j} = 1$ and $\sum_{1\le i,j\le k} a'_{i,j} = k$. On the other hand, since $q_0 \notin Q_{q_0}$, a row of $A'$ indexed by a state accessible in one step from $q_0$ does not include all nonzero elements. Since A is doubly stochastic, $\exists i, 1 \le i \le k$, $\sum_{j=1}^{k} a'_{i,j} < 1$ and $\sum_{1\le i,j\le k} a'_{i,j} < k$. This is a contradiction. □

**Corollary 2.28.** *Suppose $A$ is a doubly stochastic matrix. Then exists $k > 0$, such that $\forall i\ (A^k)_{i,i} > 0$.*

*Proof.* Consider an $m \times m$ doubly stochastic matrix $A$. By Lemma 2.27, $\forall i$ $\exists n_i > 0\ (A^{n_i})_{i,i} > 0$. Take $n = \prod_{s=1}^{m} n_s$. For every $i$, $(A^n)_{i,i} > 0$. □

**Lemma 2.29.** *If $M$ is a doubly stochastic Markov chain with a matrix $A$, then $\forall q_s, q_t\ a_{t,s} > 0 \Rightarrow q_t \to q_s$.*

*Proof.* The idea of the proof due to M. Kravtsev. $a_{t,s} > 0$ means that $q_t$ is accessible from $q_s$ in one step. We have to prove, that $q_t \to q_s$. Assume from the contrary, that $q_s$ is not accessible from $q_t$. Let $Q_{q_t} = \{q_i \mid q_t \to q_i\} = \{q_1, q_2, \dots, q_k\}$. By Lemma 2.27, $q_t \in Q_{q_t}$. As in the proof of Lemma 2.27, consider a matrix $A'$, which is a submatrix of $A$ and whose rows and columns are indexed by states in $Q_{q_t}$. Each column $j$ has to include all nonzero elements of the corresponding column of $A$. Therefore $\forall j$, $1 \leq j \leq k$, $\sum_{i=1}^{k} a'_{i,j} = 1$ and $\sum_{1 \leq i,j \leq k} a'_{i,j} = k$. On the other hand, $a_{t,s} > 0$ and $q_s \notin Q_{q_t}$, therefore a row of $A'$ indexed by $q_t$ does not include all nonzero elements. Since $A$ is doubly stochastic, $\sum_{j=1}^{k} a'_{t,j} < 1$ and $\sum_{1 \leq i,j \leq k} a'_{i,j} < k$. This is a contradiction. □

**Corollary 2.30.** *If $M$ is a doubly stochastic Markov chain with matrix $A$ and $q_s \to q_t$, then $q_s \leftrightarrow q_t$.*

*Proof.* If $q_s \to q_t$ then exists a sequence $q_{i_1}, q_{i_2}, \ldots, q_{i_k}$, such that $a_{i_1,s} > 0$, $a_{i_2,i_1} > 0, \ldots, a_{i_k,i_{k-1}} > 0$, $a_{t,i_k} > 0$. By Lemma 2.29, we get $q_t \to q_{i_k}$, $q_{i_k} \to q_{i_{k-1}}, \ldots, q_{i_2} \to q_{i_1}$, $q_{i_1} \to q_s$. Therefore $q_t \to q_s$. □

By Corollary 2.30, every doubly stochastic Markov chain does not have transient states, so it is either periodic or aperiodic, either reducible or irreducible.

## 2.2 Automata

In this section, we define notions applicable to arbitrary type of automata in a quasi-formal way. In further chapters, these notions easily transform into formal definitions applicable to automata discussed there.

### Abstract Automaton

Consider an abstract automaton $A = (Q, \Sigma_1, \ldots, \Sigma_m, q_0, \delta)$, where $Q$ is a finite set of states, $\Sigma_k$ is an alphabet of the k-th tape, $q_0$ is the initial state and $\delta$ is a transition function. (See Figure 2.1.)

Each tape is potentially infinite on both directions. The cells of each tape are indexed by numbers in $\mathbb{Z}$. Each cell of the k-th tape stores a symbol in $\Sigma_k$ or white space, denoted $\lambda$. A cell the k-th tape head is above is called *the k-th current cell.* The transition function determines possible transitions of the automaton depending on its current configuration.

**Definition 2.31.** *A* configuration *of an abstract automaton is $c = (q_i, n_1, \sigma_1, \tau_1, \ldots, n_m, \sigma_m, \tau_m)$, where the automaton is in a state $q_i \in Q$ and $\sigma_k\tau_k \in \Sigma_k^*$ is a finite word on the k-th input tape. The k-th current cell is indexed by $n_k$ and it contains the last symbol of the word $\sigma_k$, if $\sigma_k \neq \epsilon$ and $\lambda$, otherwise. All cells before or after $\sigma_k\tau_k$ are blank (contain $\lambda$).*

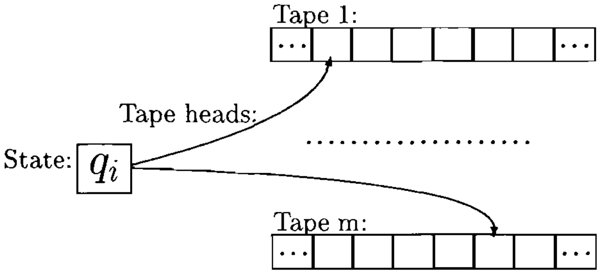

Figure 2.1: An abstract automaton

The automaton operates in discreet time moments $(t_0, \ldots, t_r, \ldots)$. If the automaton cannot change contents of a particular tape, it is called *input tape*. Let us assume that the automaton has $p$ input tapes, and renumber the tapes, so that first come input tapes. At the time moment $t_0$, the automaton is in configuration $(q_0, 0, \epsilon, \tau_1, \ldots, 0, \epsilon, \tau_p, 0, \epsilon, \epsilon, \ldots, 0, \epsilon, \epsilon)$, where $\tau_1, \ldots, \tau_p$ are *input words*. We refer to the input word tuple as *input*. At each time moment, the automaton performs a single transition, called *step*. At each step, depending on its current state and symbols in current cells, the automaton may change its current state, change the contents of current cells, and afterwards, move each tape head one cell forward or backward.

Formally, the transition function $\delta$ defines a binary relation $\rho$ from the set $Q \times \Sigma_1 \times \ldots \times \Sigma_m$ to the set $Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m$. $(q_1, s_1, \ldots, s_m)\rho(q_2, s'_{p+1}, \ldots, s'_m, d_1, \ldots, d_m)$, $d_i \in \{\leftarrow, \downarrow, \rightarrow\}$, means that for the automaton being in the state $q_1$ and having symbols $s_1, \ldots, s_m$ in current cells, the following transition is possible: the automaton goes to the state $q_2$, writes $s'_{p+1}, \ldots, s'_m$ into the current cells of the tapes $p+1, \ldots, m$ and moves tape heads according to the directions $d_i$. If this relation is a function, we speak about *deterministic automata*, other considered possibilities are *probabilistic automata* and *quantum automata*. Probabilistic automata perform transitions with certain probabilities, whereas quantum automata - with certain *amplitudes*.

For technical reasons, we may introduce two categories of white spaces for input tapes, called *end-markers*; one is used before input word and denoted as #, and the other after input word and denoted as \$. So every input word is enclosed into end-marker symbols # and \$[1]. Therefore we introduce a *working alphabet* of the k-th input tape as $\Gamma_k = \Sigma_k \cup \{\#, \$\}$. We define the

---

[1] To get rid of infinite input tapes we may also assume that input tapes are circular and the length of every input tape is $l = \max\limits_{0<k\leq p} \{|\tau_k|\} + 2$, so that the next cell after the cell indexed by $l-1$ is the cell indexed by 0. The cells indexed by 0 store # and the rest blank cells store \$.

*length of input* as the length of the longest word in the input word tuple (including one end-marker to the left of the word and one to the right of the word).

By **C** we denote the set of all configurations of an automaton. This set is countably infinite.

**Remark 2.32.** It is possible to reach only a finite number of other configurations from a given configuration in one step, all the same, within one step the given configuration is reachable only from a finite number of different configurations.

An abstract automaton introduced above is actually a description of an m-tape Turing machine. To define other types of automata, we apply specific restrictions to this general model. We say that an automaton is *1-way*, if at each step, it must move each input tape head one cell forward. We say that an automaton is *1.5-way*, if at each step, it may not move input tape heads backward. Otherwise, an automaton is called *2-way*. We refer to an automaton as a *finite automaton*, if all of its tapes are input tapes.

To halt computation of the automaton, we may consider at least two options. According to the first option, a subset of **C** is introduced and configurations in the subset are marked as *halting* configurations. We monitor the computation of the automaton and stop the computation as soon as the automaton enters a halting configuration. According to the second option, we determine the number of steps of computation in advance, and run the automaton the specified number of steps. In particular, when the number of steps is equal to the length of input, we get *real-time* automata.

## Word Acceptance

We study automata in terms of formal languages they recognize. At least two definitions exist, how to interpret word acceptance, and hence, language recognition, for automata.

**Definition 2.33.** "Decide and halt" acceptance. *Consider an automaton with the set of configurations partitioned into non-halting configurations and halting configurations, where halting configurations are further classified as accepting configurations and rejecting configurations. We say that an automaton accepts (rejects) an input in a decide-and-halt manner, if the following conditions hold:*

- *the computation is halted as soon as the automaton enters a halting configuration;*

- *if the automaton enters an accepting configuration, the input is accepted;*
- *if the automaton enters a rejecting configuration, the input is rejected.*

We refer to the decide-and-halt automata as DH-automata further in the thesis. In case of real-time automata, we may use the following definition.

**Definition 2.34.** Classical acceptance. *Consider an automaton with the set of configurations partitioned into accepting configurations and rejecting configurations. We say that an automaton accepts (rejects) an input classically, if the following conditions hold:*

- *the computation is halted as soon as the number of computation steps is equal to the length of input;*
- *if the automaton has entered an accepting configuration when halted, the input is accepted;*
- *if the automaton has entered a rejecting configuration when halted, the input is rejected.*

We refer to the classical acceptance automata as classical automata or C-automata further in the thesis.

The both definitions generally are not equivalent.

## Language Recognition

Having defined word acceptance, we define language recognition in an equivalent way as in [R 63].

By $p_{x,A}$ we denote the probability that an input $x$ is accepted by an automaton $A$.

Furthermore, we denote $P_L = \{p_{x,A} \mid x \in L\}$, $\overline{P_L} = \{p_{x,A} \mid x \notin L\}$, $p_1 = \sup \overline{P_L}$, $p_2 = \inf P_L$.

**Definition 2.35.** *We say that an automaton $A$ recognizes a language $L$ with interval $(p_1, p_2)$, if $p_1 \leq p_2$ and $P_L \cap \overline{P_L} = \emptyset$.*

**Definition 2.36.** *We say that an automaton $A$ recognizes a language $L$ with bounded error and interval $(p_1, p_2)$, if $p_1 < p_2$.*

We consider only bounded error language recognition in this thesis.

**Definition 2.37.** *An automaton recognizes a language with probability $p$ if the automaton recognizes the language with interval $(1-p, p)$.*

**Definition 2.38.** *We say that a language is recognized by some class of automata with probability* $1-\varepsilon$*, if for every* $\varepsilon > 0$ *there exists an automaton in the class which recognizes the language with interval* $(\varepsilon_1, 1-\varepsilon_2)$*, where* $\varepsilon_1, \varepsilon_2 \leq \varepsilon$.

## Quantum Automata

In case of a quantum automaton, the transition function is

$$\delta : (Q \times \Sigma_1 \times \ldots \times \Sigma_m) \times (Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m) \longrightarrow \mathbb{C}_{[0,1]}.$$

On each computation step, the quantum automaton is in quantum superposition of configurations[2] $|\psi\rangle = \sum\limits_{c\in\mathbf{C}} \alpha_c|c\rangle$, where $\sum\limits_{c\in\mathbf{C}} |\alpha_c|^2 = 1$ and $\alpha_c \in \mathbb{C}$ is the amplitude of a configuration $|c\rangle$. Every configuration $|c\rangle \in \mathbf{C}$ is a basis vector in the Hilbert space $H$, determined by $l_2(\mathbf{C})$. Every quantum automaton defines a linear operator (evolution) over this Hilbert space. Due to the laws of quantum mechanics, this operator must be unitary. Although evolution operator matrix is infinite, by Remark 2.32 it has a finite number of nonzero elements in each row and column, therefore it is possible to derive necessary and sufficient conditions, i.e., *well-formedness conditions* to check unitarity for each particular automata type.

**General measurements.** After each step, a *measurement* is applied to the current quantum superposition of configurations. A measurement is defined as follows. We introduce a set partition of $\mathbf{C}$ as $\{\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C_z}\}$. So $\bigcup\limits_{0<i\leq z} \mathbf{C_i} = \mathbf{C}$ and if $i \neq j$ then $\mathbf{C_i} \cap \mathbf{C_j} = \emptyset$. $E_1, E_2, \ldots, E_z$ are subspaces of $H$ spanned by $\mathbf{C_1}, \mathbf{C_2}, \ldots, \mathbf{C_z}$, respectively. We use the observable $\mathcal{O}_1$ that corresponds to the orthogonal decomposition $H = E_1 \oplus E_2 \oplus \ldots \oplus E_z$. If the quantum superposition before the observation is $\sum\limits_{c\in\mathbf{C}} \alpha_c|c\rangle$, with probability $p_i = \sum\limits_{c\in\mathbf{C_i}} |\alpha_c|^2$ the outcome of the observation is $|\psi_i\rangle = \frac{1}{\sqrt{p_i}} \sum\limits_{c\in\mathbf{C_i}} \alpha_c|c\rangle$. Hence the total outcome of the observation is a *mixed state* $\sum\limits_{i=1}^{z} p_i|\psi_i\rangle\langle\psi_i|$.

If $z = 1$, we get quantum automata with pure states, otherwise we generally have quantum automata with mixed states. We get other marginal case, when $\mathbf{C}$ is set partitioned into infinitely many subsets, with a single configuration in each subset[3]. In that case, the resulting quantum automaton is a

---

[2] More precisely, the automaton with certain probabilities is one of several possible quantum superpositions, or in a mixed state.

[3] By Remark 2.32, on each computation step the number of configurations in a quantum superposition is finite, so on each step it is possible to make the corresponding measurement actually using some finite partition of $\mathbf{C}$.

special kind of a probabilistic automaton. See the next subsection for further details.

**Word acceptance measurements.** Another type of measurement is applied to the quantum automaton to facilitate language recognition.

Decide-and-halt acceptance. We have to monitor when the quantum automaton enters a halting configuration. Hence we perform the following measurement after each step. We partition $\mathbf{C}$ as $\mathbf{C_a}$, $\mathbf{C_r}$ and $\mathbf{C_{non}}$, i.e., accepting, rejecting and non-halting configurations. $E_a$, $E_r$ and $E_{non}$ are subspaces of $H$ spanned by $\mathbf{C_a}$, $\mathbf{C_r}$, and $\mathbf{C_{non}}$, respectively. We use the observable $\mathcal{O}_2$ that corresponds to the orthogonal decomposition $H = E_a \oplus E_r \oplus E_{non}$. The outcome of each observation is either "accept" or "reject" or "continue". If the quantum superposition before the observation is $\sum_{c \in \mathbf{C}} \alpha_c |c\rangle$, with probability $p_a = \sum_{c \in \mathbf{C_a}} |\alpha_c|^2$ the input is accepted, with probability $p_r = \sum_{c \in \mathbf{C_r}} |\alpha_c|^2$ the input is rejected, and with probability $p_{non} = \sum_{c \in \mathbf{C_{non}}} |\alpha_c|^2$ the automaton is in the quantum superposition of non-halting states $|\psi\rangle = \frac{1}{\sqrt{p_{non}}} \sum_{c \in \mathbf{C_{non}}} \alpha_c |c\rangle$.

Classical acceptance. After the computation is halted, we have to determine, whether the automaton has entered accepting or rejecting configuration. We partition $\mathbf{C}$ as $\mathbf{C_{acc}}$ and $\mathbf{C_{rej}}$, i.e., accepting and rejecting configurations. $E_{acc}$, $E_{rej}$ are subspaces of $H$ spanned by $\mathbf{C_{acc}}$ and $\mathbf{C_{rej}}$, respectively. We use the observable $\mathcal{O}_3$ that corresponds to the orthogonal decomposition $H = E_{acc} \oplus E_{rej}$. The outcome of the observation is either "accept" or "reject". If the quantum superposition before the observation is $\sum_{c \in \mathbf{C}} \alpha_c |c\rangle$, with probability $p_{acc} = \sum_{c \in \mathbf{C_{acc}}} |\alpha_c|^2$ the input is accepted and with probability $p_{rej} = \sum_{c \in \mathbf{C_{rej}}} |\alpha_c|^2$ the input is rejected.

In case both general measurement and word acceptance measurement have to be performed in a single step, it is easy to see that the order of measurements is irrelevant, actually both measurements may be combined into a single measurement after each step.

Putting things together, each computation step consists of two parts. At first the unitary evolution operator is applied to the current quantum superposition and then the appropriate measurements are applied, using observables as defined above.

## Probabilistic Reversible Automata

Let us consider A. Nayak's model of quantum automata with mixed states, [N 99]. A variety of this model for arbitrary type of automata was considered in the previous subsection. (The difference is that Nayak's model allows a fixed sequence of unitary transformations and subsequent measurements after each step.) As noted there, if a result of every observation is a single configuration, not a superposition of configurations, we actually get a probabilistic automaton. However, the following property applies to such probabilistic automata - their evolution matrices are *doubly* stochastic.

The transition function is

$$\delta : (Q \times \Sigma_1 \times \ldots \times \Sigma_m) \times (Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m) \longrightarrow \mathbb{R}_{[0,1]}.$$

After its every step, the probabilistic automaton is in some probability distribution $p_0c_0 + p_1c_1 + \ldots + p_zc_z$, where $p_0 + p_1 + \ldots + p_z = 1$. Such probability distribution is called a superposition of configurations.

A linear closure of $\mathbf{C}$ forms a linear space, where every configuration can be viewed as a basis vector. This basis is called a *canonical basis*. Every probabilistic automaton defines a linear operator (evolution) over this linear space. The corresponding evolution matrix must be doubly stochastic. So we give the following definition for probabilistic reversible automata:

**Definition 2.39.** *A probabilistic automaton is called* reversible *if its evolution is described by a doubly stochastic matrix, using canonical basis.*

If the evolution of a probabilistic reversible automaton is described by *unitary stochastic* matrix (see Definition 2.14), the automaton can be viewed as a special case of a quantum automaton with mixed states.

It is necessary to note that in [AF 98], A. Ambainis and R. Freivalds proposed a more restricted notion of probabilistic reversibility. For example, they remarked that for the language $L = \{a^{2n+3} | n \in \mathbb{N}\}$, not recognizable by a 1-way deterministic reversible finite automata, there exists a 1-way probabilistic reversible finite automaton which recognizes the language. Consequently, this restricted notion was used in [YKTI 00]. That model is actually a restricted special case of probabilistic reversible DH-automata, as defined in the thesis.

## Deterministic Reversible Automata

Deterministic reversible automata can be viewed both as a special case of quantum automata or as a special case of probabilistic reversible automata. The transition function is

$$\delta : (Q \times \Sigma_1 \times \ldots \times \Sigma_m) \times (Q \times \Sigma_{p+1} \times \ldots \times \Sigma_m \times \{\leftarrow, \downarrow, \rightarrow\}^m) \longrightarrow \{0, 1\}.$$

## Automata Notations

We regard quantum automata, probabilistic reversible automata and deterministic reversible automata as reversible automata. Refering to different types of automata, we shall use the following notation:

$$[\mathrm{C}|\mathrm{DH}\text{-}]\langle\text{automata type}\rangle[\text{-}\mathrm{P}|\mathrm{M}].$$

C refers to "classical", whereas DH refers to "decide-and-halt". Notations P and M are used in the case of quantum automata. P denotes an automaton with pure states, whereas M - an automaton with mixed states.

For example, C-QFA-M are quantum finite automata with mixed states, using classical definition of language recognition.

# Chapter 3

# Probabilistic Reversible Finite Automata

## 3.1 1-way Probabilistic Reversible C-Automata

**Definition 3.1.** *1-way probabilistic reversible C-automaton (C-PRA)* $A = (Q, \Sigma, q_0, Q_F, \delta)$ *is specified by a finite set of states* $Q$*, a finite input alphabet* $\Sigma$*, an initial state* $q_0 \in Q$*, a set of accepting states* $Q_F \subseteq Q$*, and a transition function*

$$\delta : Q \times \Gamma \times Q \longrightarrow \mathbb{R}_{[0,1]},$$

*where* $\Gamma = \Sigma \cup \{\#, \$\}$ *is the input tape alphabet of* $A$ *and* $\#$, $\$$ *are end-markers not in* $\Sigma$*. Furthermore, transition function satisfies the following requirements:*

$$\forall (q_1, \sigma_1) \in Q \times \Gamma \ \sum_{q \in Q} \delta(q_1, \sigma_1, q) = 1 \tag{3.1}$$

$$\forall (q_1, \sigma_1) \in Q \times \Gamma \ \sum_{q \in Q} \delta(q, \sigma_1, q_1) = 1 \tag{3.2}$$

For every input symbol $\sigma \in \Gamma$, the transition function may be determined by a $|Q| \times |Q|$ matrix $V_\sigma$, where $(V_\sigma)_{i,j} = \delta(q_j, \sigma, q_i)$.

**Lemma 3.2.** *All matrices* $V_\sigma$ *are doubly stochastic iff conditions (3.1) and (3.2) of Definition 3.1 hold.*

*Proof.* Trivial. □

We define word acceptance as specified in Definition 2.34. The set of rejecting states is $Q \setminus Q_F$. We define language recognition as in Definition 2.36.

A linear operator $U_A$ corresponds to the automaton $A$. Formal definition of this operator follows:

**Definition 3.3.** *Given a configuration* $c = \langle \nu_i q_j \sigma \nu_k \rangle$,

$$U_A c \overset{\text{def}}{=} \sum_{q \in Q} \delta(q_j, \sigma, q) \langle \nu_i \sigma q \nu_k \rangle.$$

*Given a superposition of configurations* $\psi = \sum_{c \in C} p_c c$,

$$U_A \psi \overset{\text{def}}{=} \sum_{c \in C} p_c U_A c.$$

Using canonical basis, $U_A$ is described by an infinite matrix $M_A$.

To comply with Definition 2.39, we have to state the following:

**Lemma 3.4.** *Matrix* $M_A$ *is doubly stochastic iff conditions (3.1) and (3.2) of Definition 3.1 hold.*

*Proof.* Condition (3.1) takes place if and only if the sum of elements in every column in $M_A$ equal to 1. Condition (3.2) takes place if and only if the sum of elements in every row in $M_A$ equal to 1. □

This completes our formal definition of C-PRA.

Use of end-markers does not affect computational power of C-PRA. For every C-PRA with end-markers which recognizes some language it is possible to construct a C-PRA without end-markers which recognizes the same language. (Number of states needed may increase, however.) See Section 3.2 for further details.

By [R 63], bounded error probabilistic automata recognize only regular languages. Hence C-PRA recognize only regular languages.

**Theorem 3.5.** *If a language is recognized by a C-PRA, it is recognized by C-PRA with probability* $1 - \varepsilon$.

*Proof.* We assume that a language $L$ is recognized by a C-PRA automaton $A = (Q, \Sigma, q_0, Q_F, \delta)$ with interval $(p_1, p_2)$. Let $\delta = \frac{1}{2}(p_1 + p_2)$.

Let us consider a system of $m$ copies of the automaton $A$, denoted as $A_m$. Let our system accept a word if more than $m\delta$ automata in the system have accepted the word, and otherwise reject the word. We define language recognition by the system as in Definition 2.36.

Let us consider a word $\omega \in L$. The automaton $A$ accepts $\omega$ with probability $p_\omega \geq p_2$. As a result of reading $\omega$, $\mu_m^\omega$ automata of the system accept the word, and the rest reject it. The system accepts the word, if $\frac{\mu_m^\omega}{m} > \delta$. Let us take $\eta_0$, such that $0 < \eta_0 < p_2 - \delta \le p_w - \delta$. Estimating the probability that $\frac{\mu_m^\omega}{m} > \delta$, we have

$$P\left\{\frac{\mu_m^\omega}{m} > \delta\right\} \ge P\left\{p_\omega - \eta_0 < \frac{\mu_m^\omega}{m} < p_\omega + \eta_0\right\} = P\left\{\left|\frac{\mu_m^\omega}{m} - p_\omega\right| < \eta_0\right\} \tag{3.3}$$

In case of $m$ Bernoulli trials, Chebyshev's inequality may be used to prove the following ([GS 97], p. 312):

$$P\left\{\left|\frac{\mu_m^\omega}{m} - p_\omega\right| \ge \eta_0\right\} \le \frac{p_\omega(1-p_\omega)}{m\eta_0^2} \le \frac{1}{4m\eta_0^2} \tag{3.4}$$

The last inequality induces that

$$P\left\{\left|\frac{\mu_m^\omega}{m} - p_\omega\right| < \eta_0\right\} \ge 1 - \frac{1}{4m\eta_0^2} \tag{3.5}$$

Finally, putting (3.3) and (3.5) together,

$$P\left\{\frac{\mu_m^\omega}{m} > \delta\right\} \ge 1 - \frac{1}{4m\eta_0^2} \tag{3.6}$$

Inequality (3.6) is true for every $\omega \in L$.

On the other hand, let us consider a word $\xi \notin L$. The automaton $A$ accepts $\xi$ with probability $p_\xi \le p_1$. If we take the same $\eta_0$, $0 < \eta_0 < \delta - p_1 \le \delta - p_\xi$ and for every $\xi$ we have

$$P\left\{\frac{\mu_m^\xi}{m} > \delta\right\} \le P\left\{\left|\frac{\mu_m^\xi}{m} - p_\xi\right| \ge \eta_0\right\} \le \frac{1}{4m\eta_0^2} \tag{3.7}$$

Due to (3.6) and (3.7), for every $\varepsilon > 0$, if we take $n > \frac{1}{4\varepsilon\eta_0^2}$, we get a system $A_n$ which recognizes the language $L$ with interval $(\varepsilon_1, 1-\varepsilon_2)$, where $\varepsilon_1, \varepsilon_2 < \varepsilon$.

Let us show that $A_n$ can be simulated by a C-PRA. The automaton $A' = (Q', \Sigma, q_0', Q_F', \delta')$ is constructed as follows:

$Q' \overset{\text{def}}{=} \{\langle q_{s_1} q_{s_2} \dots q_{s_n}\rangle \mid 0 \le s_i \le |Q|-1\}$; $q_0' \overset{\text{def}}{=} \langle q_0 q_0 \dots q_0\rangle$.

A sequence $\langle q_{s_1} q_{s_2} \dots q_{s_n}\rangle$ is an accepting state of $A'$ if more than $n\delta$ elements in the sequence are accepting states of $A$. We have defined the set $Q_F'$.

Given $\sigma \in \Gamma$, $\delta'(\langle q_{a_1} q_{a_2} \dots q_{a_n}\rangle, \sigma, \langle q_{b_1} q_{b_2} \dots q_{b_n}\rangle) \overset{\text{def}}{=} \prod_{i=1}^{n} \delta(q_{a_i}, \sigma, q_{b_i})$.

In essence, $Q'$ is n-th Cartesian power of $Q$ and the linear space formed by $A'$ is n-th tensor power of the linear space formed by $A$. If we take a symbol $\sigma \in \Gamma$, transition is determined by $|Q|^n \times |Q|^n$ matrix $V'_\sigma$, which is n-th matrix direct power of $V_\sigma$, i.e, $V'_\sigma = \bigotimes_{i=1}^{n} V_\sigma$.

$A'$ simulates the system $A_n$. By Lemma 2.12, matrix direct product of two doubly stochastic matrices is a doubly stochastic matrix, so $\forall\sigma$ $V'_\sigma$ are doubly stochastic matrices. Therefore our automaton $A'$ is a C-PRA.

We have proved that $\forall\varepsilon > 0$ the language $L$ is recognized by some C-PRA with interval $(\varepsilon_1, 1-\varepsilon_2)$, where $\varepsilon_1, \varepsilon_2 < \varepsilon$. Therefore the language $L$ is recognized with probability $1-\varepsilon$. □

**Lemma 3.6.** *If a language is recognized by a C-PRA $A$ with interval $(p_1, p_2)$, exists a C-PRA which recognizes the language with probability $p$, where*

$$p = \begin{cases} \frac{p_2}{p_1+p_2}, & \textit{if } p_1+p_2 \geq 1 \\ \frac{1-p_1}{2-p_1-p_2}, & \textit{if } p_1+p_2 < 1. \end{cases}$$

*Proof.* Let us assume, that the automaton $A$ has $n-1$ states. We consider the case $p_1+p_2 \geq 1$.

Informally, having read end-marker symbol #, we simulate the automaton $A$ with probability $\frac{1}{p_1+p_2}$ and reject input with probability $\frac{p_1+p_2-1}{p_1+p_2}$.

Formally, to recognize the language with probability $\frac{p_2}{p_1+p_2}$, we modify the automaton $A$. We add a new state $q_r \notin Q_F$, and change the transition function in the following way:

- $\forall\sigma$, $\sigma \neq \#$, $\delta(q_r, \sigma, q_r) \overset{\text{def}}{=} 1$;
- $\delta(q_0, \#, q_r) \overset{\text{def}}{=} \frac{p_1+p_2-1}{p_1+p_2}$;
- $\forall q$, $q \neq q_r$, $\delta(q_0, \#, q) \overset{\text{def}}{=} \frac{1}{p_1+p_2}\delta_{old}(q_0, \#, q)$.

Now the automaton has $n$ states. Since end-marker symbol # is read only once at the beginning of an input word, we can disregard the rest of transition function values, associated with #: $\forall q_i, q_j$, where $q_i \neq q_0$, $\delta(q_i, \#, q_j) \overset{\text{def}}{=} \frac{1-\delta(q_0,\#,q_j)}{n-1}$.

The transition function satisfies the requirements of Definition 3.1 and the constructed automaton recognizes the language with probability $\frac{p_2}{p_1+p_2}$.

The case $p_1+p_2 < 1$ is very similar. Informally, having read end-marker symbol #, we simulate the automaton $A$ with probability $\frac{1}{2-p_1-p_2}$ and accept input with probability $\frac{1-p_1-p_2}{2-p_1-p_2}$. □

**Lemma 3.7.** *If a language* $L_1$ *is recognizable with probability greater than* $\frac{2}{3}$ *and a language* $L_2$ *is recognizable with probability greater than* $\frac{2}{3}$ *then languages* $L_1 \cap L_2$ *and* $L_1 \cup L_2$ *are recognizable with probability greater than* $\frac{1}{2}$.

*Proof.* Let us consider automata $A = (Q_A, \Sigma, q_{0,A}, Q_{F,A}, \delta_A)$ and $B = (Q_B, \Sigma, q_{0,B}, Q_{F,B}, \delta_B)$ which recognize the languages $L_1$, $L_2$ with probabilities $p_1, p_2 > \frac{2}{3}$, respectively. Let us assume that $A, B$ have $m$ and $n$ states, respectively. Without loss of generality we can assume that $p_1 \leq p_2$.

Informally, having read end-marker symbol #, with probability $\frac{1}{2}$ we simulate the automaton $A_1$ and with the same probability we simulate the automaton $A_2$.

Formally, we construct an automaton $C = (Q, \Sigma, q_0, Q_F, \delta)$ with the following properties.
$Q \overset{\text{def}}{=} Q_A \cup Q_B$; $q_0 \overset{\text{def}}{=} q_{0,A}$; $Q_F \overset{\text{def}}{=} Q_{F,A} \cup Q_{F,B}$; $\delta \overset{\text{def}}{=} \delta_A \cup \delta_B$, with an exception that:

- $\delta(q_0, \#, q_{i,A}) = \frac{1}{2}\delta_A(q_0, \#, q_{i,A})$;
- $\delta(q_0, \#, q_{i,B}) = \frac{1}{2}\delta_B(q_0, \#, q_{i,B})$;
- $\forall q_i,\ q_i \neq q_0,\ \delta(q_i, \#, q) = \frac{1-\delta(q_0,\#,q)}{m+n-1}$.

Since $\delta$ satisfies Definition 3.1, our construction of C-PRA is complete.

The automaton $C$ recognizes the language $L_1 \cap L_2$ with interval $(1-a_1, b_1)$, where $a_1 \geq \frac{1}{2}p_1$, $b_1 \geq \frac{p_1+p_2}{2}$. (Since $p_1, p_2 > \frac{2}{3}$, $1 - a_1 < b_1$.)

The automaton $C$ recognizes the language $L_1 \cup L_2$ with interval $(1-b_2, a_2)$, where $a_2 \geq \frac{1}{2}p_1$, $b_2 \geq \frac{p_1+p_2}{2}$. (Again, $1 - b_2 < a_2$.)

Therefore by Lemma 3.6, the languages $L_1 \cap L_2$ and $L_1 \cup L_2$ are recognizable with probabilities greater than $\frac{1}{2}$. □

**Theorem 3.8.** *The class of languages recognized by C-PRA is closed under intersection, union and complement.*

*Proof.* Let us consider languages $L_1, L_2$ recognized by some C-PRA automata. By Theorem 3.5, these languages is recognizable with probability $1 - \varepsilon$, and therefore by Lemmas 3.6 and 3.7, union and intersection of these languages are also recognizable. If a language $L$ is recognizable by a C-PRA $A$, we can construct an automaton which recognizes a language $\overline{L}$ just by making accepting states of $A$ to be rejecting, and vice versa. □

It is natural to ask what are the languages recognized by C-PRA with probability exactly 1.

**Theorem 3.9.** *If a language is recognized by a C-PRA with probability* 1, *the language is recognized by a permutation automaton.*

*Proof.* Let us consider a language $L$ and a C-PRA $A$, which recognizes $L$ with probability 1.

If a word is in $L$, the automaton $A$ has to accept the word with probability 1. Conversely, if a word is not in $L$, the word must be accepted with probability 0. Therefore,

$$\forall q \in Q \ \forall \omega \in \Sigma^* \text{ either } q\omega \subseteq Q_F, \text{ or } q\omega \subseteq \overline{Q}_F. \tag{3.8}$$

Consider a relation between the states of $A$ defined as

$R = \{(q_i, q_j) \mid \forall \omega \ q_i\omega \subseteq Q_F \Leftrightarrow q_j\omega \subseteq Q_F\}$. $R$ is symmetric, reflexive and transitive, therefore $Q$ can be partitioned into equivalence classes $Q/R = \{[q_0], [q_{i_1}], \ldots, [q_{i_k}]\}$. Suppose $A$ is in a state $q$. Due to (3.8), $\forall \omega \ \exists n \ q\omega \subseteq [q_{i_n}]$. In fact, having read a symbol in the alphabet, $A$ goes from one equivalence class to another with probability 1.

Hence it is possible to construct the following deterministic automaton $D$, which simulates $A$. The states are $s_0, \ldots, s_k$ and $s_n\sigma = s_m$ iff $[q_{i_n}]\sigma \subseteq [q_{i_m}]$ and $s_n$ is an accepting state iff $[q_{i_n}] \subseteq Q_F$. Since all transition matrices of $A$ are doubly stochastic, all transition matrices of $D$ are permutation matrices. □

**Theorem 3.10.** *The class of languages recognized by C-PRA is closed under inverse homomorphisms.*

*Proof.* Let us consider finite alphabets $\Sigma, T$, a homomorphism $h : \Sigma \longrightarrow T^*$, a language $L \subseteq T^*$ and a C-PRA $A = (Q, T, q_0, Q_F, \delta)$, which recognizes $L$ with interval $(p_1, p_2)$. We prove that exists an automaton $B = (Q, \Sigma, q_0, Q_F, \delta')$ which recognizes the language $h^{-1}(L)$.

Transition function $\delta$ of A sets transition matrices $V_\tau$, where $\tau \in T$. To determine $\delta'$, we define transition matrices $V_\sigma$, $\sigma \in \Sigma$. Let us define a transition matrix $V_{\sigma_k}$:

$$V_{\sigma_k} = V_{[h(\sigma_k)]_m} V_{[h(\sigma_k)]_{m-1}} \cdots V_{[h(\sigma_k)]_1},$$

where $m = |h(\sigma_k)|$. Multiplication of two doubly stochastic matrices is a doubly stochastic matrix, therefore $B$ is a C-PRA. Automaton $B$ recognizes $h^{-1}(L)$ with interval $(a_1, a_2)$, where $a_1 \leq p_1$, $a_2 \geq p_2$. □

**Corollary 3.11.** *The class of languages recognized by C-PRA is closed under word quotient.*

*Proof.* This follows from closure under inverse homomorphisms and presence of end-markers $\#, \$$. □

Even if C-PRA without end-markers are considered, closure under word quotient remains true. (See Section 3.2.)

**Theorem 3.12.** *For every natural positive $n$, a language $L_n = a_1^* a_2^* \ldots a_n^*$ is recognizable by some C-PRA with alphabet $\{a_1, a_2, \ldots, a_n\}$.*

*Proof.* We construct a C-PRA with $n+1$ states, $q_0$ being the initial state, corresponding to probability distribution vector $\begin{pmatrix} 1 \\ 0 \\ \ldots \\ 0 \end{pmatrix}$. The transition function is determined by $(n+1) \times (n+1)$ matrices

$$V_{a_1} = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & \frac{1}{n} & \ldots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{1}{n} & \ldots & \frac{1}{n} \end{pmatrix}, V_{a_2} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & \ldots & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & \ldots & 0 \\ 0 & 0 & \frac{1}{n-1} & \ldots & \frac{1}{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \frac{1}{n-1} & \ldots & \frac{1}{n-1} \end{pmatrix}, \ldots,$$

$$V_{a_n} = \begin{pmatrix} \frac{1}{n} & \ldots & \frac{1}{n} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{1}{n} & \ldots & \frac{1}{n} & 0 \\ 0 & \ldots & 0 & 1 \end{pmatrix}.$$

The accepting states are $q_0 \ldots q_{n-1}$, the only rejecting state is $q_n$. We prove, that the automaton recognizes the language $L_n$.

Case $\omega \in L_n$. Having read $\omega \in a_1^* \ldots a_{k-1}^* a_k^+$, the automaton is in probability distribution $\begin{pmatrix} \frac{1}{k} \\ \ldots \\ \frac{1}{k} \\ 0 \\ \ldots \\ 0 \end{pmatrix}$. Therefore all $\omega \in L_n$ are accepted with probability 1.

Case $\omega \notin L_n$. Consider $k$ such that $\omega = \omega_1 \sigma \omega_2$, $|\omega_1| = k$, $\omega_1 \in L_n$ and $\omega_1 \sigma \notin L_n$. Since all one-letter words are in $L_n$, $k > 0$. Let $a_t = [\omega]_k$ and $a_s = \sigma$. So we have $1 \leq s < t \leq n$. Having read $\omega_1 \in a_1^* \ldots a_{t-1}^* a_t^+$, the automaton is in distribution $\begin{pmatrix} \frac{1}{t} \\ \dots \\ \frac{1}{t} \\ 0 \\ \dots \\ 0 \end{pmatrix}$. After that, having read $a_s$, the automaton is in distribution

$$\begin{pmatrix} \frac{1}{s} & \dots & \frac{1}{s} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{1}{s} & \dots & \frac{1}{s} & 0 & \dots & 0 \\ 0 & \dots & 0 & \frac{1}{n-s+1} & \dots & \frac{1}{n-s+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \frac{1}{n-s+1} & \dots & \frac{1}{n-s+1} \end{pmatrix} \begin{pmatrix} \frac{1}{t} \\ \dots \\ \frac{1}{t} \\ 0 \\ \dots \\ 0 \end{pmatrix} = \begin{pmatrix} \left.\begin{matrix} \frac{1}{t} \\ \dots \\ \frac{1}{t} \end{matrix}\right\} s \\ \left.\begin{matrix} \frac{t-s}{t(n-s+1)} \\ \dots \\ \frac{t-s}{t(n-s+1)} \end{matrix}\right\} n-s+1 \end{pmatrix}.$$

So the word $\omega_1 a_s$ is accepted with probability $1 - \frac{t-s}{t(n-s+1)}$. By Lemma 2.13, since $\frac{t-s}{t(n-s+1)} < \frac{1}{t}$, reading the symbols succeeding $\omega_1 a_s$ does not increase accepting probability. Therefore, to find maximum accepting probability for words not in $L_n$, we have to maximize $1 - \frac{t-s}{t(n-s+1)}$, where $1 \leq s < t \leq n$. Solving this problem, we get $t = k+1, s = k$ for $n = 2k$, and we get $t = k+1, s = k$ or $t = k+2, s = k+1$ for $n = 2k+1$. So the maximum accepting probability is $1 - \frac{1}{(k+1)^2}$, if $n = 2k$, and it is $1 - \frac{1}{(k+1)(k+2)}$, if $n = 2k+1$. All in all, the automaton recognizes the language with interval $\left(1 - \frac{1}{\lfloor (\frac{n}{2})^2 \rfloor + n + 1}, 1\right)$. (Actually, by Theorem 3.5, $L_n$ can be recognized with probability $1 - \varepsilon$). □

**Corollary 3.13.** *Quantum finite automata with mixed states (model of Nayak, [N 99]) recognize* $L_n = a_1^* a_2^* \dots a_n^*$ *with probability* $1 - \varepsilon$.

*Proof.* This comes from the fact, that matrices $V_{a_1}, V_{a_2}, \dots, V_{a_n}$ from the proof of Theorem 3.12 (as well as direct powers of those matrices) are unitary stochastic (see Lemma 2.4, Definition 2.14, Theorem 3.30). □

**Definition 3.14.** *We say that a regular language is of Type* 0 *(Figure 3.1) if the following is true for the minimal deterministic automaton recognizing this language: Exist three states* $q$, $q_1$, $q_2$, *exist words* $x$, $y$ *such that*

1) $q_1 \neq q_2$;
2) $qx = q_1$, $qy = q_2$;
3) $q_1 x = q_1$, $q_2 y = q_2$;
4) $\forall t \in (x,y)^* \ \exists t_1 \in (x,y)^* \ q_1 t t_1 = q_1$;
5) $\forall t \in (x,y)^* \ \exists t_2 \in (x,y)^* \ q_2 t t_2 = q_2$.

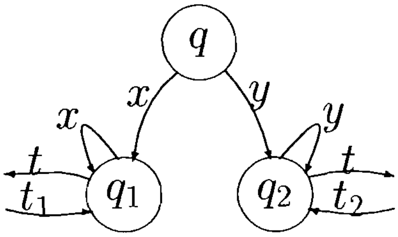

Figure 3.1: Type 0 construction

**Definition 3.15.** *We say that a regular language is of Type* 1 *(Figure 3.2) if the following is true for the minimal deterministic automaton recognizing this language: Exist two states* $q_1$, $q_2$, *exist words* $x$, $y$ *such that*
1) $q_1 \neq q_2$;
2) $q_1x = q_2$, $q_2x = q_2$;
3) $q_2y = q_1$.

**Definition 3.16.** *We say that a regular language is of Type* 2 *(Figure 3.3) if the following is true for the minimal deterministic automaton recognizing this language: Exist three states* $q$, $q_1$, $q_2$, *exist words* $x$, $y$ *such that*
1) $q_1 \neq q_2$;
2) $qx = q_1$, $qy = q_2$;
3) $q_1x = q_1$, $q_1y = q_1$;
4) $q_2x = q_2$, $q_2y = q_2$.

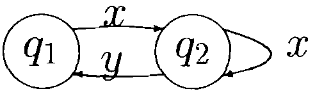

Figure 3.2: Type 1 construction

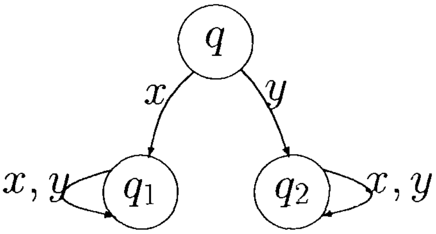

Figure 3.3: Type 2 construction

Type 1 languages are exactly those languages that violate the partial order condition of [BP 99].

**Lemma 3.17.** *If* $A$ *is a deterministic finite automaton with a set of states* $Q$ *and alphabet* $\Sigma$, *then* $\forall q \in Q\ \forall x \in \Sigma^*\ \exists k > 0\ qx^k = qx^{2k}$.

*Proof.* We paraphrase a result from the theory of finite semigroups. Consider a state $q$ and a word $x$. Since number of states is finite, $\exists m \geq 0\ \exists s \geq 1\ \forall n$ $qx^m = qx^m x^{sn}$. Take $n_0$, such that $sn_0 > m$. Note that $\forall t \geq 0\ qx^{m+t} = qx^{m+t}x^{sn_0}$. We take $t = sn_0 - m$, so $qx^{sn_0} = qx^{sn_0}x^{sn_0}$. Take $k = sn_0$. □

**Lemma 3.18.** *A regular language is of Type* 0 *iff it is of Type* 1 *or Type* 2.

*Proof.* 1) If a language is of Type 2, it is of Type 0. Obvious.

2) If a language is of Type 1, it is of Type 0. Consider a language of Type 1 with states $q_1'', q_2''$ and words $x'', y''$. To build construction of Type 0, we take $q = q_1 = q_1''$, $q_2 = q_2''$, $x = x''y''$, $y = x''$. That forms transitions $qx = q_1$, $qy = q_2$, $q_1x = q_1$, $q_1y = q_2$, $q_2x = q_1$, $q_2y = q_2$. We have satisfied all the rules of Type 0.

3) If a language is of Type 0, it is of Type 1 or 2. Consider a language whose minimal deterministic automaton has construction of Type 0. By Lemma 3.17,

$$\exists t \exists b \; q_1 y^b = q_t \text{ and } q_t y^b = q_t;$$
$$\exists u \exists c \; q_2 x^c = q_u \text{ and } q_u x^c = q_u.$$

If $q_1 \neq q_t$, by the 4th rule of Type 0, $\exists z \; q_t z = q_1$. Therefore the language is of Type 1. If $q_2 \neq q_u$, by the 5th rule of Type 0, $\exists z \; q_u z = q_2$, and the language is of Type 1.

If $q_1 = q_t$ and $q_2 = q_u$, we have $qx^c = q_1$, $qy^b = q_2$, $q_1x^c = q_1y^b = q_1$, $q_2x^c = q_2y^b = q_2$. We get the construction of Type 2 if we take $x' = x^c$, $y' = y^b$. □

The following theorem illustrates the relationship between Type 1 and Type 2 languages.

**Theorem 3.19.** *A regular language $L$ is of Type* 1 *iff $L^R$ is of Type* 2.

*Proof.* It is a well known fact, that the class of regular languages is closed under reversal.

1) Consider a Type 1 regular language $L \subset \Sigma^*$. Since L is of Type 1, it is recognized by a minimal deterministic automaton $D = (Q, \Sigma, q_0, Q_F, \delta)$ with particular two states $q_1, q_2$, such that $q_1 \neq q_2$, $q_1x = q_2$, $q_2x = q_2$, $q_2y = q_1$, where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0\omega = q_1$, and exists $z \in \Sigma^*$ such that $q_1z \in Q_F$ if and only if $q_2z \notin Q_F$. Minimal deterministic automata of a regular language and of its complement are isomorphic, so without loss of generality we assume that $q_1z \in Q_F$ and $q_2z \notin Q_F$.

So $\omega\{xy, x\}^*xz \subset \overline{L}$ and $\omega\{xy, x\}^*(xy)z \subset L$, and in the case of the reverse of $L$, $z^Rx^R\{y^Rx^R, x^R\}^*\omega^R \subset \overline{L^R}$ and $z^R(y^Rx^R)\{y^Rx^R, x^R\}^*\omega^R \subset L^R$. We denote $\sigma_1 = x^R$, $\sigma_2 = y^Rx^R$, hence $z^R\sigma_1\{\sigma_2, \sigma_1\}^*\omega^R \subset \overline{L^R}$ and $z^R\sigma_2\{\sigma_2, \sigma_1\}^*\omega^R \subset L^R$.

Consider a minimal deterministic automaton $D^R = (Q^R, \Sigma, s_0, Q_F^R, \delta^R)$, which recognizes $L^R$. Let $s = s_0z^R$. Let $Q_1 = \{s\tau \mid \tau \in \sigma_1\{\sigma_2, \sigma_1\}^*\}$ and $Q_2 = \{s\tau \mid \tau \in \sigma_2\{\sigma_2, \sigma_1\}^*\}$. For any $q \in Q_1$, $q\omega^R \notin Q_F^R$ and for any $q \in Q_2$,

$q\omega^R \in Q_F^R$. Therefore $Q_1 \cap Q_2 = \emptyset$. Furthermore, it is impossible to go from a state in $Q_1$ to a state in $Q_2$, or vice versa, using only words in $\{\sigma_1, \sigma_2\}^*$. So $s \notin Q_1$ and $s \notin Q_2$.

Consider a relation $R = \{(s_i, s_j) \in Q_1^2 \mid s_j \in s_i\{\sigma_1, \sigma_2\}^*\}$. $R$ is a weak ordering, so $R' = \{(s_i, s_j) \mid s_i R s_j \text{ and } s_j R s_i\}$ is an equivalence relation, partitioning $Q_1$ into equivalence classes. Since the number of states in $Q_1$ is finite, exists a class $S \subset Q_1$, which is minimal, i.e, $\forall q \in S\ \forall \tau \in \{\sigma_1, \sigma_2\}^*$ $q\tau \in S$. Since $S \subset Q_1$, exists a word $\tau_1 \in \{\sigma_1, \sigma_2\}^*$, such that $s(\sigma_1\tau_1) \in S$. Now by Lemma 3.17, $\exists p > 0\ \exists s_1 \in S\ s(\sigma_1\tau_1)^p = s_1$ and $s_1(\sigma_1\tau_1)^p = s_1$. Since $S$ is an equivalence class of $R'$, $\forall q \in S\ \forall \tau \in \{\sigma_1, \sigma_2\}^*\ \exists \tau_2 \in \{\sigma_1, \sigma_2\}^*$ $q(\tau\tau_2) = q$. So, exists $\tau_2$, such that $s_1(\sigma_2\tau_2) = s_1$.

Let us denote $\alpha = (\sigma_1\tau_1)^p$, $\beta = \sigma_2\tau_2$, so $s\alpha = s_1$, $s_1\alpha = s_1$, $s_1\beta = s_1$, where $s_1 \in Q_1$.

By Lemma 3.17, it is possible to construct a sequence of states $t_0$, $t_1, \ldots,$ $t_{m-1}, \ldots,$ where $t_0 = s$, such that

$t_0(\beta\alpha^{k_1}) = t_1$ and $t_1\alpha^{k_1} = t_1$,

$t_1(\beta\alpha^{k_2}) = t_2$ and $t_2\alpha^{k_2} = t_2$,

...

$t_{m-1}(\beta\alpha^{k_m}) = t_m$ and $t_m\alpha^{k_m} = t_m$,

...

Because $\beta \in \sigma_2\{\sigma_1, \sigma_2\}^*$ and $\alpha \in \sigma_1\{\sigma_1, \sigma_2\}^*$, $\forall i > 0\ t_i \in Q_2$. Let $T_m = \{t_0, \ldots, t_m\}$. Since the number of states in $Q_2$ is finite, exists $i$, such that $t_i \in T_{i-1}$. So, exists $j$, $0 < j < i$, such that $t_j = t_i$ and starting with $t_j$, the sequence becomes periodic. Let $k = k_1k_2\ldots k_i$. Now, $\forall m \geq 0\ t_m(\beta\alpha^k) = t_{m+1}$ and $t_{m+1}\alpha^k = t_{m+1}$. By Lemma 3.17, $\exists r > 0\ \exists s_2$, such that $s(\beta\alpha^k)^r = s_2$ and $s_2(\beta\alpha^k)^r = s_2$. The state $s_2 = t_r$, so $s_2 \in Q_2$ and $s_2\alpha^k = s_2$.

So we have $s\alpha^k = s_1$, $s_1\alpha^k = s_1$, $s_1(\beta\alpha^k)^r = s_1$, $s(\beta\alpha^k)^r = s_2$, $s_2(\beta\alpha^k)^r = s_2$, $s_2\alpha^k = s_2$. Since $s_1 \in Q_1$, $s_2 \in Q_2$, $s_1$ is not equal to $s_2$, thus we have obtained a Type 2 construction.

2) Consider a Type 2 regular language $L \subset \Sigma^*$. Since L is of Type 2, it is recognized by a minimal deterministic automaton $D = (Q, \Sigma, q_0, Q_F, \delta)$ with particular three states $q$, $q_1$, $q_2$, such that $q_1 \neq q_2$, $qx = q_1$, $q_1x = q_1$, $q_1y = q_1$, $qy = q_2$, $q_2x = q_2$, $q_2y = q_2$, where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0\omega = q$, and exists $z \in \Sigma^*$ such that $q_1z \in Q_F$ if and only if $q_2z \notin Q_F$. Without loss of generality we assume that $q_1z \in Q_F$ and $q_2z \notin Q_F$.

So $\omega x\{x, y\}^*z \subset L$ and $\omega y\{x, y\}^*z \subset \overline{L}$, and in the case of the reverse of $L$, $z^R\{x^R, y^R\}^*x^R\omega^R \subset L^R$ and $z^R\{x^R, y^R\}^*y^R\omega^R \subset \overline{L^R}$. We denote $\sigma_1 = x^R$, $\sigma_2 = y^R$, hence $z^R\{\sigma_1, \sigma_2\}^*\sigma_1\omega^R \subset L^R$ and $z^R\{\sigma_1, \sigma_2\}^*\sigma_2\omega^R \subset \overline{L^R}$.

Consider a minimal deterministic automaton $D^R = (Q^R, \Sigma, s_0, Q_F^R, \delta^R)$, which recognizes $L^R$. Let $s = s_0z^R$. Let $Q_1 = \{s\tau \mid \tau \in \{\sigma_1, \sigma_2\}^*\sigma_1\}$ and

$Q_2 = \{s\tau \mid \tau \in \{\sigma_1, \sigma_2\}^*\sigma_2\}$. For any $t \in Q_1$, $t\omega^R \in Q_F^R$ and for any $t \in Q_2$, $t\omega^R \notin Q_F^R$. Therefore $Q_1 \cap Q_2 = \emptyset$.

Let $T = Q_1 \cup Q_2$. Consider a relation $R = \{(s_i, s_j) \in T^2 \mid s_j \in s_i\{\sigma_1, \sigma_2\}^*\}$. $R$ is a weak ordering, so $R' = \{(s_i, s_j) \mid s_i R s_j \text{ and } s_j R s_i\}$ is an equivalence relation, partitioning $T$ into equivalence classes. Since the number of states in $T$ is finite, exists a class $S \subset T$, which is minimal, i.e, $\forall t \in S\ \forall \tau \in \{\sigma_1, \sigma_2\}^*\ t\tau \in S$.

Consider a state $t \in S$. If the state $t$ is in $Q_1$ then $t\sigma_2 \in S$ is in $Q_2$. If the state $t$ is in $Q_2$ then $t\sigma_1 \in S$ is in $Q_1$. So exist $t_1$, $t_2$, such that $t_1 \in Q_1 \cap S$, $t_2 \in Q_2 \cap S$. Take $s_1 \in Q_1 \cap S$. By Lemma 3.17, $\exists k > 0\ \exists s_2$, such that $s_1\sigma_2^k = s_2$ and $s_2\sigma_2^k = s_2$. The state $s_2$ is in $Q_2 \cap S$. Since $S$ is an equivalence class of $R'$, $\exists \sigma \in \{\sigma_1, \sigma_2\}^*$, such that $s_2\sigma = s_1$.

So we have $s_1\sigma_2^k = s_2$, $s_2\sigma_2^k = s_2$, $s_2\sigma = s_1$. Since $s_1 \in Q_1$, $s_2 \in Q_2$, $s_1$ is not equal to $s_2$, thus we have obtained a Type 1 construction. □

**Remark 3.20.** Both C-DRA and C-QFA-P (see Section 3.3) recognize exactly the regular languages for which the corresponding minimal deterministic finite automata do not contain the following construction ([HS 66, BP 99]), denoted henceforth as Type $A$ construction (Figure 3.4): Exist two states $q_1$, $q_2$, exist words $x$, $y$ such that

1) $q_1 \neq q_2$;

2) $q_1 x = q_2$, $q_2 x = q_2$.

Similarly as in Theorem 3.19, it is possible to demonstrate that a regular language L is of Type $A$ if and only if the language $L^R$ is of Type $A$.

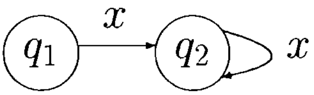

Figure 3.4: Type $A$ construction

We are going to prove that every language of Type 0 is not recognizable by any C-PRA.

**Definition 3.21.** *By* $q \xrightarrow{S} q'$, $S \subset \Sigma^*$, *we denote that there is a positive probability to get to a state* $q'$ *by reading a single word* $\xi \in S$, *starting in a state* $q$.

**Lemma 3.22.** *If a regular language is of Type* 2, *it is not recognizable by any C-PRA.*

*Proof.* Assume from the contrary, that $A$ is a C-PRA automaton which recognizes a language $L \subset \Sigma^*$ of Type 2.

Since $L$ is of Type 2, it is recognized by a minimal deterministic automaton $D$ with particular three states $q$, $q_1$, $q_2$ such that $q_1 \neq q_2$, $qx = q_1$, $qy = q_2$, $q_1x = q_1$, $q_1y = q_1$, $q_2x = q_2$, $q_2y = q_2$, where $x, y \in \Sigma^*$. Furthermore, exists $\omega \in \Sigma^*$ such that $q_0\omega = q$, where $q_0$ is an initial state of $D$, and exists a word $z \in \Sigma^*$, such that $q_1z = q_{acc}$ if and only if $q_2z = q_{rej}$, where $q_{acc}$ is an accepting state and $q_{rej}$ is a rejecting state of $D$. Without loss of generality we assume that $q_1z = q_{acc}$ and $q_2z = q_{rej}$.

The transition function of the automaton $A$ is determined by doubly stochastic matrices $V_{\sigma_1}, \dots, V_{\sigma_n}$. The words from the construction of Type 2 are $x = \sigma_{i_1} \dots \sigma_{i_k}$ and $y = \sigma_{j_1} \dots \sigma_{j_s}$. The transitions induced by words $x$ and $y$ are determined by doubly stochastic matrices $X = V_{\sigma_{i_k}} \dots V_{\sigma_{i_1}}$ and $Y = V_{\sigma_{j_s}} \dots V_{\sigma_{j_1}}$. Similarly, the transitions induced by words $\omega$ and $z$ are determined by doubly stochastic matrices $W$ and $Z$. By Corollary 2.28, exists $K > 0$, such that

$$\forall i \ (X^K)_{i,i} > 0 \text{ and } (Y^K)_{i,i} > 0. \tag{3.9}$$

Consider a relation between the states of the automaton defined as $R = \{(q_i, q_j) \mid q_i \xrightarrow{(x^K, y^K)^*} q_j\}$. By (3.9), this relation is reflexive.

Suppose exists a word $\xi = \xi_1\xi_2 \dots \xi_k$, $\xi_s \in \{x^K, y^K\}$, such that $q \xrightarrow{\xi} q'$. This means that $q \xrightarrow{\xi_1} q_{i_1}$, $q_{i_1} \xrightarrow{\xi_2} q_{i_2}, \dots, q_{i_{k-1}} \xrightarrow{\xi_k} q'$. By Corollary 2.30, since both $X^K$ and $Y^K$ are doubly stochastic, $\exists \xi'_k \dots \xi'_1$, $\xi'_s \in \{(x^K)^*, (y^K)^*\}$, such that $q' \xrightarrow{\xi'_k} q_{i_{k-1}}, \dots, q_{i_2} \xrightarrow{\xi'_2} q_{i_1}$, $q_{i_1} \xrightarrow{\xi'_1} q$, therefore $q' \xrightarrow{\xi'} q$, where $\xi' \in (x^K, y^K)^*$. So the relation $R$ is symmetric.

Surely $R$ is transitive. Therefore all states of $A$ may be partitioned into equivalence classes $[q_0], [q_{i_1}], \dots, [q_{i_n}]$. Let us renumber the states of $A$ in such a way, that states from one equivalence class have consecutive numbers. First come the states in $[q_0]$, then in $[q_{i_1}]$, etc.

Consider the word $x^Ky^K$. The transition induced by this word is determined by a doubly stochastic matrix $C = Y^KX^K$. We prove the following proposition. States $q_a$ and $q_b$ are in one equivalence class if and only if $q_a \to q_b$ with matrix $C$. Suppose $q_a \to q_b$. Then $(q_a, q_b) \in R$, and $q_a, q_b$ are in one equivalence class. Suppose $q_a, q_b$ are in one equivalence class. Then

$$q_a \xrightarrow{\xi_1} q_{i_1}, q_{i_1} \xrightarrow{\xi_2} q_{i_2}, \dots, q_{i_{k-1}} \xrightarrow{\xi_k} q_b, \text{ where } \xi_s \in \{x^K, y^K\}. \tag{3.10}$$

By (3.9), $q_i \xrightarrow{x^K} q_i$ and $q_j \xrightarrow{y^K} q_j$. Therefore, if $q_i \xrightarrow{x^K} q_j$, then $q_i \xrightarrow{x^Ky^K} q_j$, and again, if $q_i \xrightarrow{y^K} q_j$, then $q_i \xrightarrow{x^K y^K} q_j$. That transforms (3.10) to

$$q_a \xrightarrow{(x^K y^K)^t} q_b, \text{ where } t > 0. \tag{3.11}$$

We have proved the proposition.

By the proved proposition, due to the renumbering of states, matrix $C$ is a block diagonal matrix, where each block corresponds to an equivalence class of the relation $R$. Let us identify these blocks as $C_0, C_1, \ldots, C_n$. By (3.9), a Markov chain with matrix C is aperiodic. Therefore each block $C_r$ corresponds to an aperiodic irreducible doubly stochastic Markov chain with states $[q_{i_r}]$. By Corollary 2.26, $\lim\limits_{m\to\infty} C^m = J$, $J$ is a block diagonal matrix, where for each $(p \times p)$ block $C_r$ $(C_r)_{i,j} = \frac{1}{p}$. Relation $q_i \xrightarrow{(y^K)^*} q_j$ is a subrelation of $R$, therefore $Y^K$ is a block diagonal matrix with the same block ordering and sizes as $C$ and $J$. (This does not eliminate possibility that some block of $Y^K$ is constituted of smaller blocks, however.) Therefore $JY^K = J$, and $\lim\limits_{m\to\infty} Z(Y^K X^K)^m W = \lim\limits_{m\to\infty} Z(Y^K X^K)^m Y^K W = ZJW$. So

$$\forall \varepsilon > 0\ \exists m\ \left\| \left( Z(Y^K X^K)^m W - Z(Y^K X^K)^m Y^K W \right) Q_0 \right\| < \varepsilon. \tag{3.12}$$

However, by construction of Type 2, $\forall k\ \forall m\ \omega(x^k y^k)^m z \in L$ and $\omega y^k (x^k y^k)^m z \notin L$. This requires existence of $\varepsilon > 0$, such that

$$\forall m\ \left\| \left( Z(Y^K X^K)^m W - Z(Y^K X^K)^m Y^K W \right) Q_0 \right\| > \varepsilon. \tag{3.13}$$

This is a contradiction. □

**Lemma 3.23.** *If a regular language is of Type* 1, *it is not recognizable by any C-PRA.*

*Proof.* Proof is nearly identical to that of Lemma 3.22. Consider a C-PRA which recognizes the language $L$ of Type 1. We prove that for words $x$, $y$ exists constant $K$, such that for every $\varepsilon$ exists $m$, such that for two words $\xi_1 = \omega(x^K(xy)^K)^m z$ and $\xi_2 = \omega(x^K(xy)^K)^m x^K z$, $|p_{\xi_1} - p_{\xi_2}| < \varepsilon$. We can choose $z$, such that $\xi_1 \in L$ iff $\xi_2 \notin L$. □

**Theorem 3.24.** *If a regular language is of Type* 0, *it is not recognizable by any C-PRA.*

*Proof.* By Lemmas 3.18, 3.22, 3.23. □

We proved (Lemma 3.18) that the construction of Type 0 is a generalization the construction proposed by [BP 99]. Also it can be easily noticed, that the Type 0 construction is a generalization of construction proposed by [AKV 00]. (Constructions of [BP 99] and [AKV 00] characterize languages, not recognized by measure-many quantum finite automata of [KW 97].)

**Corollary 3.25.** *Languages (a,b)*a and a(a,b)* are not recognized by C-PRA.*

*Proof.* Both languages are of Type 0. □

**Corollary 3.26.** *Class of languages recognizable by C-PRA is not closed under homomorphisms.*

*Proof.* Consider a homomorphism $a \to a$, $b \to b$, $c \to a$. Similarly as in Theorem 3.12, the language (a,b)*cc* is recognizable by a C-PRA. (Take $n = 2$, $V_a = V_{a_1}$, $V_b = V_{a_1}$, $V_c = V_{a_2}$ from Theorem 3.12, $Q_F = \{q_1\}$) However, by Corollary 3.25 the language (a,b)*aa*=(a,b)*a is not recognizable. □

## 3.2 End-Marker Theorems for C-PRA

In this section, we prove that the use of end-markers in case of C-PRA is optional.

We denote a C-PRA with both end-markers as #,\$-C-PRA. We denote a C-PRA with left end-marker only as #-C-PRA.

**Theorem 3.27.** *Let A be a #,\$-C-PRA, which recognizes a language L. There exists a #-C-PRA which recognizes the same language.*

*Proof.* Suppose $A = (Q, \Sigma, q_0, Q_F, \delta)$, where $|Q| = n$. $A$ recognizes $L$ with interval $(p_1, p_2)$. We construct the following automaton $A' = (Q', \Sigma, q_{0,0}, Q'_F, \delta')$ with $mn$ states. Informally, $A'$ equiprobably simulates $m$ copies of the automaton $A$.

$Q' = \{q_{0,0}, \dots, q_{0,m-1}, q_{1,0}, \dots, q_{1,m-1}, \dots, q_{n-1,0}, \dots, q_{n-1,m-1}\}$.

If $\sigma \neq \#$, $\delta'(q_{i,k}, \sigma, q_{j,l}) = \begin{cases} \delta(q_i, \sigma, q_j), \text{ if } k = l \\ 0, \text{ if } k \neq l. \end{cases}$

Otherwise, $\delta'(q_{0,0}, \#, q_{j,l}) = \frac{1}{m}\delta(q_0, \#, q_j)$, and if $q_{i,k} \neq q_{0,0}$, $\delta'(q_{i,k}, \#, q) = \frac{1-\delta'(q_{0,0}, \#, q)}{mn-1}$. Function $\delta'$ satisfies the requirements (3.1) and (3.2) of Definition 3.1.

We define $Q'_F$ as follows. A state $q_{i,k} \in Q'_F$ if and only if $0 \leq k < mp(q_i)$, where $p(q_i) \overset{\text{def}}{=} \sum_{q \in Q_F} \delta(q_i, \$, q)$.

Suppose $\#\omega\$$ is an input word. Having read $\#\omega$, $A$ is in superposition $\sum_{i=0}^{n-1} a_i^\omega q_i$. After $A$ has read $\$$, $\#\omega\$$ is accepted with probability $p_\omega = \sum_{i=0}^{n-1} a_i^\omega p(q_i)$.

On the other hand, having read $\#\omega$, $A'$ is in superposition $\frac{1}{m}\sum_{j=0}^{m-1}\sum_{i=0}^{n-1} a_i^\omega q_{i,j}$. So the input word $\#\omega$ is accepted with probability $p'_\omega = \frac{1}{m}\sum_{i=0}^{n-1} a_i^\omega \lceil mp(q_i)\rceil$.

Consider $\omega \in L$. Then $p'_\omega = \frac{1}{m}\sum_{i=0}^{n-1} a_i^\omega \lceil mp(q_i)\rceil \geq \sum_{i=0}^{n-1} a_i^\omega p(q_i) = p_\omega \geq p_2$.

Consider $\xi \notin L$. Then $p'_\xi = \frac{1}{m}\sum_{i=0}^{n-1} a_i^\xi \lceil mp(q_i)\rceil < \sum_{i=0}^{n-1} a_i^\xi p(q_i) + \frac{1}{m}\sum_{i=0}^{n-1} a_i^\xi = p_\xi + \frac{1}{m} \leq p_1 + \frac{1}{m}$.

Therefore $A'$ recognizes $L$ with bounded error, provided $m > \frac{1}{p_2 - p_1}$. □

Now we are going to prove that C-PRA without end-markers recognize the same languages as #-C-PRA automata.

If $A$ is a #-C-PRA, then, having read the left end-marker #, the automaton simulates some other automata $A_0, A_1, \ldots, A_{m-1}$ with positive probabilities $p_0, \ldots, p_{m-1}$, respectively. $A_0, A_1, \ldots, A_{m-1}$ are automata without end-markers. By $p_{i,\omega}$, $0 \leq i < m$, we denote the probability that the automaton $A_i$ accepts the word $\omega$.

We prove the following lemma first.

**Lemma 3.28.** *Suppose $A'$ is a #-C-PRA which recognizes a language $L$ with interval $(a_1, a_2)$. Then for every $\varepsilon$. $0 < \varepsilon < 1$, exists a #-C-PRA $A$ which recognizes $L$ with interval $(a_1, a_2)$, such that*

*a) if $\omega \in L$, $p_{0,\omega} + p_{1,\omega} + \ldots + p_{n-1,\omega} > \frac{a_2 n}{1+\varepsilon}$*

*b) if $\omega \notin L$, $p_{0,\omega} + p_{1,\omega} + \ldots + p_{n-1,\omega} < \frac{a_1 n}{1-\varepsilon}$.*

*Here $n$ is the number of automata without end-markers, being simulated by $A$, and $p_{i,\omega}$ is the probability that $i$-th simulated automaton $A_i$ accepts $\omega$.*

*Proof.* Suppose a #-C-PRA $A'$ recognizes a language $L$ with interval $(a_1, a_2)$. Having read the symbol #, $A'$ simulates automata $A'_0, \ldots, A'_{m-1}$ with probabilities $p'_0, \ldots, p'_{m-1}$, respectively. We choose $\varepsilon$, $0 < \varepsilon < 1$.

By Dirichlet's principle ([HW 79], p. 170), $\forall \varphi > 0$ exists $n \in \mathbb{N}^+$ such that $\forall i$ $p'_i n$ differs from some positive integer by less than $\varphi$.

Let $0 < \varphi < \min\left(\frac{1}{m}, \varepsilon\right)$. Let $g_i$ be the nearest integer of $p'_i n$. So $|p'_i n - g_i| < \varphi$ and $\left|\frac{p'_i}{g_i} - \frac{1}{n}\right| < \frac{\varphi}{n g_i} \le \frac{\varphi}{n}$. Since $|p'_i n - g_i| < \varphi$, we have $\left|n - \sum\limits_{i=0}^{m-1} g_i\right| < \varphi m < 1$. Therefore, since $g_i \in \mathbb{N}^+$, $\sum\limits_{i=0}^{m-1} g_i = n$.

Now we construct the #-C-PRA $A$, which satisfies the properties expressed in Lemma 3.28. For every $i$, we make $g_i$ copies of $A'_i$. Having read #, for every $i$ $A$ simulates each copy of $A'_i$ with probability $\frac{p'_i}{g_i}$. The construction of $V_\#$ is equivalent to that used in the proof of Lemma 3.7. Therefore $A$ is characterized by doubly stochastic matrices. $A$ recognizes $L$ with the same interval as $A'$, i.e., $(a_1, a_2)$.

Using new notations, $A$ simulates $n$ automata $A_0, A_1, \ldots, A_{n-1}$ with probabilities $p_0, p_1, \ldots, p_{n-1}$, respectively. Note that $\forall i \ \left|p_i - \frac{1}{n}\right| < \frac{\varphi}{n}$. Let $p_{i,\omega}$ be the probability that $A_i$ accepts the word $\omega$.

Consider $\omega \in L$. We have $p_0 p_{0,\omega} + p_1 p_{1,\omega} + \ldots + p_{n-1} p_{n-1,\omega} \ge a_2$. Since $p_i < \frac{1+\varphi}{n}$, $\frac{1+\varphi}{n}(p_{0,\omega} + p_{1,\omega} + \ldots + p_{n-1,\omega}) > a_2$. Hence

$$p_{0,\omega} + p_{1,\omega} + \ldots + p_{n-1,\omega} > \frac{a_2 n}{1+\varphi} > \frac{a_2 n}{1+\varepsilon}.$$

Consider $\xi \notin L$. We have $p_0 p_{0,\xi} + p_1 p_{1,\xi} + \ldots + p_{n-1} p_{n-1,\xi} \le a_1$. Since $p_i > \frac{1-\varphi}{n}$, $\frac{1-\varphi}{n}(p_{0,\xi} + p_{1,\xi} + \ldots + p_{n-1,\xi}) < a_1$. Hence

$$p_{0,\xi} + p_{1,\xi} + \ldots + p_{n-1,\xi} < \frac{a_1 n}{1-\varphi} < \frac{a_1 n}{1-\varepsilon}.$$

□

**Theorem 3.29.** *Let $A$ be a #-C-PRA, which recognizes a language $L$. There exists a C-PRA without end-markers, which recognizes the same language.*

*Proof.* Consider a #-C-PRA which recognizes a language $L$ with interval $(a_1, a_2)$. Using Lemma 3.28, we choose $\varepsilon$, $0 < \varepsilon < \frac{a_2 - a_1}{a_2 + a_1}$, and construct an automaton $A'$ which recognizes L with interval $(a_1, a_2)$, with the following properties.

Having read #, $A'$ simulates $A'_0, \ldots, A'_{m-1}$ with probabilities $p'_0, \ldots, p'_{m-1}$, respectively. $A'_0, \ldots, A'_{m-1}$ are automata without end-markers. $A'_i$ accepts $\omega$ with probability $p'_{i,\omega}$. If $\omega \in L$, $p'_{0,\omega} + p'_{1,\omega} + \ldots + p'_{m-1,\omega} > \frac{a_2 m}{1+\varepsilon}$. Otherwise, if $\omega \notin L$, $p'_{0,\omega} + p'_{1,\omega} + \ldots + p'_{m-1,\omega} < \frac{a_1 m}{1-\varepsilon}$.

That also implies that for every $n = km$, $k \in \mathbb{N}^+$, we are able to construct a #-C-PRA $A$ which recognizes $L$ with interval $(a_1, a_2)$, such that

a) if $\omega \in L$, $p_{0,\omega} + p_{1,\omega} + \ldots + p_{n-1,\omega} > \frac{a_2 n}{1+\varepsilon}$;

b) if $\omega \notin L$, $p_{0,\omega} + p_{1,\omega} + \ldots + p_{n-1,\omega} < \frac{a_1 n}{1-\varepsilon}$.

$A$ simulates $A_0, \ldots, A_{n-1}$. Let us consider the system $F_n = (A_0, \ldots, A_{n-1})$. Let $\delta = \frac{1}{2}(a_1 + a_2)$. Since $\varepsilon < \frac{a_2 - a_1}{a_2 + a_1}$, $\frac{a_2}{1+\varepsilon} > \delta$ and $\frac{a_1}{1-\varepsilon} < \delta$. As in the proof of Theorem 3.5, we define that the system accepts a word, if more than $n\delta$ automata in the system accept the word.

Let us take $\eta_0$, such that $0 < \eta_0 < \frac{a_2}{1+\varepsilon} - \delta < \delta - \frac{a_1}{1-\varepsilon}$.

Consider $\omega \in L$. We have that $\sum_{i=0}^{n-1} p_{i,\omega} > \frac{a_2 n}{1+\varepsilon} > n\delta$. As a result of reading $\omega$, $\mu_n^\omega$ automata in the system accept the word, and the rest reject it. The system has accepted the word, if $\frac{\mu_n^\omega}{n} > \delta$. Since $0 < \eta_0 < \frac{a_2}{1+\varepsilon} - \delta < \frac{1}{n}\sum_{i=0}^{n-1} p_{i,\omega} - \delta$, we have

$$P\left\{\frac{\mu_n^\omega}{n} > \delta\right\} \geq P\left\{\left|\frac{\mu_n^\omega}{n} - \frac{1}{n}\sum_{i=0}^{n-1} p_{i,\omega}\right| < \eta_0\right\}. \tag{3.14}$$

If we look on $\frac{\mu_n^\omega}{n}$ as a random variable $X$, $E(X) = \frac{1}{n}\sum_{i=0}^{n-1} p_{i,\omega}$ and variance $V(X) = \frac{1}{n^2}\sum_{i=0}^{n-1} p_{i,\omega}(1 - p_{i,\omega})$, therefore Chebyshev's inequality yields the following:

$$P\left\{\left|\frac{\mu_n^\omega}{n} - \frac{1}{n}\sum_{i=0}^{n-1} p_{i,\omega}\right| \geq \eta_0\right\} \leq \frac{1}{n^2\eta_0^2}\sum_{i=0}^{n-1} p_{i,\omega}(1 - p_{i,\omega}) \leq \frac{1}{4n\eta_0^2}.$$

That is equivalent to $P\left\{\left|\frac{\mu_n^\omega}{n} - \frac{1}{n}\sum_{i=0}^{n-1} p_{i,\omega}\right| < \eta_0\right\} \geq 1 - \frac{1}{4n\eta_0^2}$. So, taking into account (3.14),

$$P\left\{\frac{\mu_n^\omega}{n} > \delta\right\} \geq 1 - \frac{1}{4n\eta_0^2}. \tag{3.15}$$

On the other hand, consider $\xi \notin L$. So $\sum_{i=0}^{n-1} p_{i,\xi} < \frac{a_1 n}{1-\varepsilon} < n\delta$. Again, since $0 < \eta_0 < \delta - \frac{a_1}{1-\varepsilon} < \delta - \frac{1}{n}\sum_{i=0}^{n-1} p_{i,\xi}$,

$$P\left\{\frac{\mu_n^\xi}{n} > \delta\right\} \leq P\left\{\left|\frac{\mu_n^\xi}{n} - \frac{1}{n}\sum_{i=0}^{n-1} p_{i,\xi}\right| \geq \eta_0\right\} \leq \frac{1}{4n\eta_0^2}. \tag{3.16}$$

The constant $\eta_0$ does not depend on $n$ and $n$ may be chosen sufficiently large. Therefore, by (3.15) and (3.16), the system $F_n$ recognizes $L$ with bounded error, if $n > \frac{1}{2\eta_0^2}$.

Following a way identical to that used in the proof of Theorem 3.5, it is possible to construct a single C-PRA without end-markers, which simulates the system $F_n$ and therefore recognizes the language $L$. □

## 3.3 Classification of Reversible Automata

We propose the following classification for one-way reversible finite automata:

| | C-Automata | DH-Automata |
|---|---|---|
| Deterministic Automata | Permutation Automata [HS 66, T 68] (C-DRA) | Reversible Finite Automata [AF 98] (DH-DRA) |
| Quantum Automata with Pure States | Measure-Once Quantum Finite Automata [MC 97] (C-QFA-P) | Measure-Many Quantum Finite Automata [KW 97] (DH-QFA-P) |
| Probabilistic Automata | Probabilistic Reversible C-Automata (C-PRA) | Probabilistic Reversible DH-Automata (DH-PRA) |
| Quantum Finite Automata with Mixed States | not considered in any known paper (C-QFA-M) | Enhanced Quantum Finite Automata [N 99] (DH-QFA-M) |

Language class problems have been solved for C-DRA, DH-DRA, C-QFA-P, for the remaining types they are still open. Every type of DH-automata may simulate the corresponding type of C-automata.

The following relation among language classes also presents interest, question marks denoting conjectures:

$$\text{C-DRA} = \text{C-QFA-P} \subset \text{C-PRA} \overset{?}{=} \text{C-QFA-M}$$
$$\text{DH-DRA} \subset \text{DH-QFA-P} \overset{?}{\subset} \text{DH-PRA} \overset{?}{\subset} \text{DH-QFA-M}$$

Generally, language classes recognized by C-automata are closed under boolean operations (though this is open for C-QFA-M), while DH-automata are not (though this is open for DH-QFA-M and possibly for DH-PRA).

Below Definition 2.39, we demonstrated some relation between PRA and QFA-M. However, due to Remark 2.15, we do not know exactly, whether every C-PRA can be simulated by C-QFA-M, or whether every DH-PRA can be simulated by DH-QFA-M.

The following results are however straightforward.

**Theorem 3.30.** *If all matrices of a C-PRA are unitary stochastic, then the C-PRA may be simulated by a C-QFA-M and by a DH-QFA-M.*

**Theorem 3.31.** *If all matrices of a DH-PRA are unitary stochastic, then the DH-PRA may be simulated by a DH-QFA-M.*

**Remark 3.32.** Remark 2.15 gives an example of a doubly stochastic matrix, which is not unitary stochastic, namely, $A = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$. A. Ambainis [Am 02] however noted the following. Consider a matrix $B$, which is formed from the matrix $A$ by doubling the number of rows and columns and substituting $\frac{1}{2}$ by the $2 \times 2$ submatrix consisting of $\frac{1}{4}$.

$$B = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}, U = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}.$$

The matrix $B$ is unitary stochastic, the corresponding unitary matrix is $U$. Clearly, the matrix $A$ may be simulated by the matrix $B$. Perhaps it is possible to generalize this, so that every PRA may be simulated by QFA-M.

# Chapter 4

# Quantum Finite Multitape Automata

## 4.1 Definition of QFMA

We are using these notations in the following definition:
$M \overset{\text{def}}{=} \{1, 2, \ldots, m\}$.
The $k$-th component of an arbitrary vector $s$ will be defined as $s^k$.
We shall understand by $I$ an arbitrary element from the set $\mathcal{P}(M) \setminus \{\emptyset\}$.

$R_I \overset{\text{def}}{=} A_1 \times A_2 \times \ldots \times A_m$, where $A_i = \begin{cases} \{\downarrow, \rightarrow\}, & \text{if } i \notin I \\ \{\text{“}nothing\text{”}\}, & \text{if } i \in I. \end{cases}$

$T_I \overset{\text{def}}{=} B_1 \times B_2 \times \ldots \times B_m$, where $B_i = \begin{cases} \{\downarrow, \rightarrow\}, & \text{if } i \in I \\ \{\text{“}nothing\text{”}\}, & \text{if } i \notin I. \end{cases}$

The function $R_i \times T_i \xrightarrow{d_I} \{\downarrow, \rightarrow\}^m$ is defined as follows:
$d_I(r,t) \overset{\text{def}}{=} (d_I^1(r,t), d_I^2(r,t), \ldots, d_I^m(r,t))$, where $d_I^i(r,t) = \begin{cases} r^i, & \text{if } i \notin I \\ t^i, & \text{if } i \in I. \end{cases}$

We consider quantum finite multitape automata with pure states, as described in Quantum Automata subsection of Section 2.2.

**Definition 4.1.** *A quantum finite multitape automaton (QFMA)* $A = (Q; \Sigma; \delta; q_0; Q_a; Q_r)$ *is specified by the finite input alphabet* $\Sigma$*, the finite set of states* $Q$*, the initial state* $q_0 \in Q$*, the sets* $Q_a \subset Q$*,* $Q_r \subset Q$ *of accepting and rejecting states, respectively, with* $Q_a \cap Q_r = \emptyset$*, and the transition function*

$$\delta : Q \times \Gamma^m \times Q \times \{\downarrow, \rightarrow\}^m \longrightarrow \mathbb{C}_{[0,1]},$$

*where* $m$ *is the number of input tapes,* $\Gamma = \Sigma \cup \{\#, \$\}$ *is the tape alphabet of* $A$ *and* $\#,\$$ *are endmarkers not in* $\Sigma$*, which satisfies the following conditions (of well-formedness):*

1. *Local probability condition*

$$\forall (q_1, \sigma) \in Q \times \Gamma^m \quad \sum_{(q,d) \in Q \times \{\downarrow, \rightarrow\}^m} |\delta(q_1, \sigma, q, d)|^2 = 1.$$

2. *Orthogonality of column vectors condition.*

$$\forall q_1, q_2 \in Q, q_1 \neq q_2, \forall \sigma \in \Gamma^m \quad \sum_{(q,d) \in Q \times \{\downarrow, \rightarrow\}^m} \delta^*(q_1, \sigma, q, d)\delta(q_2, \sigma, q, d) = 0.$$

3. *Separability condition.*

$$\forall I \in \mathcal{P}(M) \setminus \{\emptyset\} \ \forall q_1, q_2 \in Q$$
$$\forall \sigma_1, \sigma_2 \in \Gamma^m, \ \textit{where } \forall i \notin I \ \sigma_1^i = \sigma_2^i$$
$$\forall t_1, t_2 \in T_I, \ \textit{where } \forall j \in I \ t_1^j \neq t_2^j$$
$$\sum_{(q,r) \in Q \times R_I} \delta^*(q_1, \sigma_1, q, d_I(r, t_1))\delta(q_2, \sigma_2, q, d_I(r, t_2)) = 0.$$

States from $Q_a \cup Q_r$ are called halting states and states from $Q_{non} = Q \setminus (Q_a \cup Q_r)$ are called non halting states.

To process an input word vector $x \in (\Sigma^*)^m$ by $A$ it is assumed that the input is written on every tape $k$ with the endmarkers in the form $w_x^k = \#x^k\$\$^*$ and that every such a tape, of length $n = \max\limits_{0<k\leq m} \{|x^k|\} + 2$, is circular, i.e., the symbol to the right of the last \$ is #.

For the fixed input word vector $x$ we have defined $n \in \mathbb{N}$ to be an integer which determines the length of tapes. So for every $n$ we can define $C_n$ to be the set of all possible configurations of $A$. $|C_n| = |Q|n^m$. Every such a configuration is uniquely determined by a pair $|q, s\rangle$, where $q \in Q$ and $0 \leq s^i \leq n - 1$ specifies the position of head on the $i$-th tape.

Every computation of $A$ on an input $x$ is specified by a unitary evolution in the Hilbert space $H_{A,n} = l_2(C_n)$. Each configuration $c \in C_n$ corresponds to the basis vector in $H_{A,n}$. Therefore a global state of $A$ in the space $H_{A,n}$ has a form $\sum\limits_{c \in C_n} \alpha_c |c\rangle$, where $\sum\limits_{c \in C_n} |\alpha_c|^2 = 1$. If the input word vector is $x$ and the automaton $A$ is in its global state $|\psi\rangle = \sum\limits_{c \in C_n} \alpha_c |c\rangle$, then its further step is equivalent to the application of a linear operator $U_x^\delta$ over Hilbert space $l_2(C_n)$.

**Definition 4.2.** *The linear operator $U_x^\delta$ is defined as follows:*

$$U_x^\delta |\psi\rangle = \sum_{c \in C_n} \alpha_c U_x^\delta |c\rangle.$$

*If a configuration* $c = |q', s\rangle$, *then*

$$U_x^\delta|c\rangle = \sum_{(q,d)\in Q\times\{\downarrow,\rightarrow\}^m} \delta(q', \sigma(s), q, d)|q, \tau(s,d)\rangle,$$

*where* $\sigma(s) = (\sigma^1(s), \dots, \sigma^m(s))$, $\sigma^i(s)$ *specifies the* $s^i$*-th symbol on the* $i$*-th tape, and* $\tau(s,d) = (\tau^1(s,d), \dots, \tau^m(s,d))$,

$$\tau^i(s,d) = \begin{cases} (s^i+1) \bmod n, & \text{if } d^i = '\rightarrow' \\ s^i, & \text{if } d^i = '\downarrow'. \end{cases}$$

**Lemma 4.3.** *The well-formedness conditions are satisfied iff for any input* $x$ *the mapping* $U_x^\delta$ *is unitary.*

**Definition 4.4.** *We shall say that an automaton is a deterministic reversible finite multitape automaton (RFMA), if it is a QFMA with* $\delta(q_1, \sigma, q, d) \in \{0, 1\}$.

**Definition 4.5.** *A QFMA* $A = (Q; \Sigma; \delta; q_0; Q_a; Q_r)$ *is simple if for each* $\sigma \in \Gamma^m$ *there is a linear unitary operator* $V_\sigma$ *over the inner-product space* $l_2(Q)$ *and a function* $D : Q \longrightarrow \{\downarrow, \rightarrow\}^m$, *such that*

$$\forall q_1 \in Q \ \forall \sigma \in \Gamma^m \quad \delta(q_1, \sigma, q, d) = \begin{cases} \langle q|V_\sigma|q_1\rangle, & \text{if } D(q) = d \\ 0, & \text{otherwise.} \end{cases}$$

**Lemma 4.6.** *If an automaton* $A$ *is simple, then conditions of well-formedness are satisfied iff for every* $\sigma$ $V_\sigma$ *is unitary.*

We shall deal only with simple deterministic and quantum multitape automata further in the chapter.

## 4.2 Language Recognition by QFMA

Word acceptance is defined as in Definition 2.33. For each input $x$ with the corresponding integer $n$ and a QFMA $A = (Q; \Sigma; \delta; q_0; Q_a; Q_r)$ we define $C_n^a = \{(q,s)|(q,s) \in C_n, q \in Q_a\}$, $C_n^r = \{(q,s)|(q,s) \in C_n, q \in Q_r\}$, $C_n^{non} = C_n \setminus (C_n^a \cup C_n^r)$. $E_a, E_r, E_{non}$ are the subspaces of $l_2(C_n)$ spanned by $C_n^a, C_n^r, C_n^{non}$ respectively. We use the observable $\mathcal{O}$ that corresponds to the orthogonal decomposition $l_2(C_n) = E_a \oplus E_r \oplus E_{non}$. The outcome of each observation is either "accept" or "reject" or "non-halting".

For an $x \in (\Sigma^*)^m$ we consider as the input $\omega_x$, $\omega_x^k = \#x^k\$^{n-k-1}$, and assume that the computation starts with $A$ being in the configuration $|q_0, \{0\}^m\rangle$. Each computation step consists of two parts. At first the linear operator $U_{\omega_x}^\delta$

is applied to the current global state and then the resulting superposition, i.e., global state, is observed using the observable $\mathcal{O}$ as defined above. If the global state before the observation is $\sum\limits_{c \in C_n} \alpha_c|c\rangle$, then the probability that the subspace $E_i$, $i \in \{a, r, non\}$, will be chosen is $\sum\limits_{c \in C_n^i} |\alpha_c|^2$. The computation continues until the result of an observation is "accept" or "reject".

We define language recognition as in Definition 2.37.

**Definition 4.7.** *We say that a language $L$ is [m,n]-deterministically recognizable if there are $n$ simple deterministic automata $A_1$, $A_2$, ..., $A_n$ such that:*
*a) if the input is in the language $L$, then all $n$ automata $A_1$, ..., $A_n$ accept the input;*
*b) if the input is not in the language $L$, then at most $m$ of the automata $A_1$, ..., $A_n$ accept the input.*

**Definition 4.8.** *We say that a language $L$ is [m,n]-reversibly recognizable if there are $n$ simple deterministic reversible automata $A_1$, $A_2$, ..., $A_n$ such that:*
*a) if the input is in the language $L$, then all $n$ automata $A_1$, ..., $A_n$ accept the input;*
*b) if the input is not in the language $L$, then at most $m$ of the automata $A_1$, ..., $A_n$ accept the input.*

**Lemma 4.9.** *If a language $L$ is [1,n]-deterministically recognizable by 2-tape finite automata, then $L$ is recognizable by a probabilistic 2-tape finite automaton with probability $\frac{n}{n+1}$.*

*Proof.* The idea of the proof due to R. Freivalds. The probabilistic automaton starts by choosing a random integer $1 \le r \le (n+1)$. After that, if $r \le n$, then the automaton goes on simulating the deterministic automaton $A_r$, and, if $r = n+1$, then the automaton rejects the input. The inputs in $L$ are accepted with probability $\frac{n}{n+1}$, and the inputs not in the language are rejected with a probability not less than $\frac{n}{n+1}$. □

**Lemma 4.10.** *If a language $L$ is [1,n]-reversibly recognizable by 2-tape finite automata, then $L$ is recognizable by a quantum 2-tape finite automaton with probability $\frac{n}{n+1}$.*

*Proof.* The idea of the proof due to R. Freivalds. In essence the algorithm is the same as in Lemma 4.9. The automaton starts by taking $n+1$ different actions with amplitudes $\frac{1}{\sqrt{n+1}}$. (By Lemma 2.3, it is possible to construct a unitary matrix to make such a choice feasible.) After that the automaton simultaneously goes on simulating all the deterministic reversible automata $A_r$, $1 \leq r \leq (n+1)$, where the automaton $A_{n+1}$ rejects an input. The simulation of each deterministic reversible automaton uses its own non-halting, accepting and rejecting states. (Hence the probabilities are totaled, not the amplitudes.) □

First, we discuss the following 2-tape language

$$L_1 = \{(x_1 \nabla x_2, y) \mid x_1 = x_2 = y\},$$

where the words $x_1, x_2, y$ are unary.

**Lemma 4.11.** *For arbitrary natural n, the language $L_1$ is [1,n]-deterministically recognizable.*

*Proof.* Proved by R. Freivalds [Fr 78].

The language $L$ can be recognized by the following team of deterministic 1-way 2-tape finite automata $\{A_1, A_2, \ldots, A_n\}$.

The automaton $A_r$ performs cycles, each one consisting in reading $n+1$ digits from $x_1$ and $r$ digits from $y$. When the symbol $\nabla$ is met, the automaton memorizes the remainder of $x_1$ modulo $n$ and goes on (in cycles) reading $n+1$ digits from $x_2$ and $n+1-r$ digits from $y$. If the input pair of words is in the language, the processing of the two tapes takes the same time. In this case the automaton accepts the pair, otherwise the automaton rejects it. This way, the automaton accepts the pair of words if and only if there are nonnegative integers $u, v$ such that:

$$(n+1)u \leq x_1$$
$$(n+1)(u+1) > x_1$$
$$(n+1)v \leq x_2$$
$$(n+1)(v+1) > x_2$$
$$x_1 - (n+1)u = x_2 - (n+1)v = y - ru - (n+1-r)v$$

If $x_1 = x_2$, then the number $-ru - (n+1-r)v$ does not depend on the choice of $r$. Either all $x_i$ match the $y$, or no one does. If $x_1 \neq x_2$, then the numbers $-ru - (n+1-r)v$ are all different for different values of $r$. Hence at most one of them can match $y$. □

**Theorem 4.12.** *The language $L_1$ can be recognized with arbitrary probability $1 - \varepsilon$ by a probabilistic 2-tape finite automaton but this language cannot be recognized by a deterministic 2-tape finite automaton.*

*Proof.* By Lemma 4.11 $L$ is [1,n]-deterministically recognizable for arbitrary $n$. By Lemma 4.9, the language is recognizable with probability $\frac{n}{n+1}$. ☐

We wish to prove a quantum counterpart of Theorem 4.12. We need a lemma to this goal.

**Lemma 4.13.** *For arbitrary natural $n$, the language $L_1$ is $[1,n]$-reversibly recognizable.*

*Proof.* The idea of the proof due to R. Freivalds. By Lemma 4.11, the language $L_1$ is $[1,n]$-deterministically recognizable. However it is easy enough to make the construction of the automata $A_1,\dots,A_n$ in the following manner:
a) every automaton is reversible;
b) if a word pair is in the language $L_1$, then every automaton consumes the same number of steps to accept the word pair.
The last requirement will be essential further in the chapter.
If at least the first requirement is met, then the language is $[1,n]$-reversibly recognizable. ☐

**Theorem 4.14.** *The language $L_1$ can be recognized with probability $1-\varepsilon$ by a quantum 2-tape finite automaton.*

*Proof.* By Lemmas 4.10 and 4.13. ☐

**Corollary 4.15.** *Exists a multitape language, recognized by quantum finite multitape automaton, where the projection of the language to one of the tapes is not regular.*

*Proof.* Consider the language $L_1$. ☐

In an attempt to construct a 2-tape language recognizable by a quantum 2-tape finite automaton but not by probabilistic 2-tape finite automata we consider a similar language

$$L_2 = \{(x_1\nabla x_2\nabla x_3, y) \mid \text{there are exactly 2 values of } x_1,x_2,x_3 \text{ such that they equal } y\},$$

where the words $x_1,x_2,x_3,y$ are unary.

**Theorem 4.16.** *A quantum automaton exists which recognizes the language $L_2$ with a probability $\frac{9}{16}-\varepsilon$ for arbitrary positive $\varepsilon$.*

*Proof.* This automaton takes the following actions with the following amplitudes:

a) $\frac{\sqrt{3}}{4} \cdot 1$ – compares $x_1 = x_2 = y$,
b) $\frac{\sqrt{3}}{4} \cdot (\cos\frac{2\pi}{3} + i\sin\frac{2\pi}{3})$ – compares $x_2 = x_3 = y$,
c) $\frac{\sqrt{3}}{4} \cdot (\cos\frac{4\pi}{3} + i\sin\frac{4\pi}{3})$ – compares $x_1 = x_3 = y$,
d) $\frac{\sqrt{7}}{4}$ – says "accept".

By Theorem 4.14 comparison in actions a), b), c) can be accomplished. By construction in Lemma 4.10 the comparison in each action a), b), c) is implemented by starting $n+1$ different branches. Therefore in any action $i)$, $i \in \{a, b, c\}$, if a comparison is successful, the automaton will come respectively into non-halting states $q_{a,1}, \ldots, q_{a,n}$, $q_{b,1}, \ldots, q_{b,n}$, $q_{c,1}, \ldots, q_{c,n}$, reaching the symbol pair (\$, \$) on the tapes. The transition (\$, \$) for every $k = 1, \ldots, n$ is as follows:

| | $q_{a,k}$ | $q_{b,k}$ | $q_{c,k}$ |
|---|---|---|---|
| $q_{a1,k}$ | $\frac{1}{\sqrt{3}}$ | $\frac{1}{\sqrt{3}}$ | $\frac{1}{\sqrt{3}}$ |
| $q_{r,k}$ | $\frac{1}{\sqrt{3}}$ | $\frac{1}{\sqrt{3}}(\cos\frac{4\pi}{3} + i\sin\frac{4\pi}{3})$ | $\frac{1}{\sqrt{3}}(\cos\frac{2\pi}{3} + i\sin\frac{2\pi}{3})$ |
| $q_{a2,k}$ | $\frac{1}{\sqrt{3}}$ | $\frac{1}{\sqrt{3}}(\cos\frac{2\pi}{3} + i\sin\frac{2\pi}{3})$ | $\frac{1}{\sqrt{3}}(\cos\frac{4\pi}{3} + i\sin\frac{4\pi}{3})$ |

Here $q_{a1,k}, q_{a2,k}$ are accepting states and $q_{r,k}$ are rejecting states. If $y$ equals all 3 words $x_1, x_2, x_3$, then it is possible to ensure that it takes the same time to reach the end-marking symbol pair in every action on every branch. Therefore the input is accepted with probability $\frac{7}{16} + \varepsilon$ (since the amplitudes of the actions a), b), c) total to 0). If $y$ equals 2 out of 3 words $x_1, x_2, x_3$, then the input is accepted with probability $\frac{9}{16} - \varepsilon$. If $y$ equals at most one of the words $x_1, x_2, x_3$, then the input is accepted with probability $\frac{7}{16} + \varepsilon$ (only if the action d) is taken). □

Unfortunately, the following theorem holds.

**Theorem 4.17.** *A probabilistic automaton exists which recognizes the language $L_2$ with a probability $\frac{21}{40}$.*

*Proof.* The idea of the proof due to R. Freivalds. The probabilistic automaton with probability $\frac{1}{2}$ takes an action $A$ or $B$:

A) Choose a random $j$ and compare $x_j = y$. If **yes**, accept with probability $\frac{19}{20}$. If **no**, accept with probability $\frac{1}{20}$.
B) Choose a random pair $j, k$ and compare $x_j = x_k = y$. If **yes**, reject. If **no**, accept with probability $\frac{12}{20}$.

If $y$ equals all 3 words $x_1, x_2, x_3$ and the action $A$ is taken, then the input is accepted with relative probability $\frac{19}{20}$. If $y$ equals all 3 words $x_1, x_2, x_3$, then and the action $A$ is taken, then the input is accepted with relative probability

0. This gives the acceptance probability in the case if $y$ equals all 3 words $x_1, x_2, x_3$, to be $\frac{19}{40}$ and the probability of the correct result "no" to be $\frac{21}{40}$.

If $y$ equals 2 words out of $x_1, x_2, x_3$ and the action $A$ is taken, then the input is accepted with relative probability $\frac{13}{20}$. If $y$ equals 2 words out of $x_1, x_2, x_3$ and the action $B$ is taken, then the input is accepted with relative probability $\frac{8}{20}$. This gives the acceptance probability in the case if $y$ equals 2 words out of $x_1, x_2, x_3$, to be $\frac{21}{40}$.

If $y$ equals only 1 word out of $x_1, x_2, x_3$ and the action $A$ is taken, then the input is accepted with relative probability $\frac{7}{20}$. If $y$ equals only 1 word out of $x_1, x_2, x_3$ and the action $B$ is taken, then the input is accepted with relative probability $\frac{12}{20}$. This gives the acceptance probability in the case if $y$ equals only 1 word out of $x_1, x_2, x_3$, to be $\frac{19}{40}$ and the probability of the correct result "no" to be $\frac{21}{40}$.

If $y$ equals no word of $x_1, x_2, x_3$ and the action $A$ is taken, then the input is accepted with relative probability $\frac{1}{20}$. If $y$ equals no word of $x_1, x_2, x_3$ and the action $B$ is taken, then the input is accepted with relative probability $\frac{12}{20}$. This gives the acceptance probability in the case if $y$ equals no word of $x_1, x_2, x_3$, to be $\frac{13}{40}$ and the probability of the correct result "no" to be $\frac{27}{40}$. □

Now we consider a modification of the language $L_2$ which might be more difficult for a probabilistic recognition:

$$L_3 = \{(x_1 \nabla x_2 \nabla x_3, y_1 \nabla y_2) \mid \text{there is exactly one value } k \text{ such that there are exactly two values } j \text{ such that } x_j = y_k\}$$

**Theorem 4.18.** *A quantum finite 2-tape automaton exists which recognizes the language $L_3$ with a probability $\frac{12}{23} - \varepsilon$ for arbitrary positive $\varepsilon$.*

*Proof.* This automaton takes the following actions with the following amplitudes:

a) With amplitude $\sqrt{\frac{2}{23}} \times (\cos\frac{0\pi}{6} + i\sin\frac{0\pi}{6})$ compares whether $x_1 = x_2 = y_1$,

b) With amplitude $\sqrt{\frac{2}{23}} \times (\cos\frac{2\pi}{6} + i\sin\frac{2\pi}{6})$ compares whether $x_2 = x_3 = y_1$,

c) With amplitude $\sqrt{\frac{2}{23}} \times (\cos\frac{4\pi}{6} + i\sin\frac{4\pi}{6})$ compares whether $x_1 = x_3 = y_1$,

d) With amplitude $\sqrt{\frac{2}{23}} \times (\cos\frac{6\pi}{6} + i\sin\frac{6\pi}{6})$ compares whether $x_1 = x_2 = y_2$,

e) With amplitude $\sqrt{\frac{2}{23}} \times (\cos\frac{8\pi}{6} + i\sin\frac{8\pi}{6})$ compares whether $x_2 = x_3 = y_2$,

f) With amplitude $\sqrt{\frac{2}{23}} \times (\cos\frac{10\pi}{6} + i\sin\frac{10\pi}{6})$ compares whether $x_1 = x_3 = y_2$,

g) With amplitude $\sqrt{\frac{11}{23}}$ says "accept".

By Theorem 4.14 comparison in actions a), b), c), d), e), f) can be accomplished. By construction in Lemma 4.10 the comparison in each action a), b), c), d), e), f) is implemented by starting $n+1$ different branches. Therefore in any action $i$), $i \in \{a, b, c, d, e, f\}$, if a comparison is successful, the automaton will come respectively into non-halting states $q_{a,1}, \dots, q_{a,n}$, $q_{b,1}, \dots, q_{b,n}$, $q_{c,1}, \dots, q_{c,n}$, $q_{d,1}, \dots, q_{d,n}$, $q_{e,1}, \dots, q_{e,n}$, $q_{f,1}, \dots, q_{f,n}$, reaching the symbol pair $(\$, \$)$ on the tapes. The transition $(\$, \$)$ for every $k = 1, \dots, n$ is as follows:

| | $q_{a,k}$ | $q_{b,k}$ | $q_{c,k}$ | $q_{d,k}$ | $q_{e,k}$ | $q_{f,k}$ |
|---|---|---|---|---|---|---|
| $q_{a1,k}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}$ |
| $q_{r1,k}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{2\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{4\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{6\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{8\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{10\pi i}{6}}$ |
| $q_{a2,k}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{4\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{8\pi i}{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{4\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{8\pi i}{6}}$ |
| $q_{r2,k}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{6\pi i}{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{6\pi i}{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{6\pi i}{6}}$ |
| $q_{a3,k}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{8\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{4\pi i}{6}}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{8\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{4\pi i}{6}}$ |
| $q_{r3,k}$ | $\frac{1}{\sqrt{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{10\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{8\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{6\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{4\pi i}{6}}$ | $\frac{1}{\sqrt{6}}e^{\frac{2\pi i}{6}}$ |

Here $q_{a1,k}$, $q_{a2,k}$, $q_{a3,k}$ are accepting states and $q_{r1,k}$, $q_{r2,k}$, $q_{r3,k}$ are rejecting states.

If both $y_1$ and $y_2$ equal all 3 words $x_1, x_2, x_3$, then it is possible to ensure that it takes the same time to reach the end-marking symbol pair in every action on every branch. Therefore the input is accepted with probability $\frac{11}{23} + \varepsilon$ (since the amplitudes of the actions a), b), c), d), e), f) total to 0, and the automaton goes to the states $q_{r3,k}$).

If both $y_1$ and $y_2$ are mutually equal to exactly two words $x_i, x_j$, then it is possible to ensure that it takes the same time to reach the end-marking symbol pair for the actions where $x_i$ and $x_j$ are compared. The total of amplitudes for the acceptance is 0 since the amplitude for comparison of $y_1$ with arbitrary pair $x_i, x_j$ is (minus 1) times the amplitude for the comparison of $y_2$ with the same pair $x_i, x_j$. The automatons goes to the states $q_{r1,k}$, $q_{r2,k}$, $q_{r3,k}$. So the input is accepted with probability $\frac{11}{23} + \varepsilon$.

If $y_1 \neq y_2$, and $y_1 = x_1 = x_2$, then $y_2$ cannot equal more than one of the $x_j$, namely, $x_3$. In this case, all the branches in action a) end in acception and so do also no more than one of the branches in action b), no more than one of the branches in action c), no more than one of the branches in action d), no more than one of the branches in action e) and no more than one of the branches in action f). So the input is accepted with probability $\frac{12}{23} - \varepsilon$. $\square$

However this language also can be recognized by a probabilistic 2-tape finite automaton.

**Theorem 4.19.** *A probabilistic finite 2-tape automaton exists which recognizes the language $L_3$ with a probability $\frac{13}{25} - \varepsilon$ for arbitrary positive $\varepsilon$.*

*Proof.* The idea of the proof due to R. Freivalds. The probabilistic automaton with probability $\frac{6}{25}$ takes action $A$ or $B$ or $C$ or with probability $\frac{7}{25}$ takes action $D$:
A) Choose a random $k$ and two values of $j$. Then compare $x_j = y_k$. If **yes**, accept. If **no**, reject.
B) Chose a random $k$ and compare $x_1 = x_2 = x_3 = y_k$. If **yes**, reject. If **no**, accept. C) Choose two values $j$ and $m$. Then compare $x_j = x_m = y_1 = y_2$. If **yes**, reject. If **no**, accept.
D) Says "reject".

Notice that the actions $A, B, C$ are probabilistic, and they can be performed only with probability $1 - \varepsilon$ (actions $A$ and $B$ are described in the proof of Theorem 4.12 and action $C$ is similar).

The acceptance probabilities equal:

| | A | B | C | Total |
|---|---|---|---|---|
| no $y_k$ equals 2 or 3 $x_j$ | 0 | 1 | 1 | $\frac{12}{25}$ |
| one $y_k$ equals 2 $x_j$ | $\frac{1}{6}$ | 1 | 1 | $\frac{13}{25}$ |
| one $y_k$ equals 3 $x_j$ | $\frac{1}{2}$ | $\frac{1}{2}$ | 1 | $\frac{12}{25}$ |
| two $y_k$ equal 2 $x_j$ | $\frac{1}{3}$ | 1 | $\frac{2}{3}$ | $\frac{12}{25}$ |
| all $y_k$ equal all $x_j$ | 1 | 0 | 0 | $\frac{6}{25}$ |

□

Finally we consider a modification of the languages above which recognition indeed looks impossible for probabilistic automata:

$$L_4 = \{(x_1 \nabla x_2, y) \mid \text{there is exactly one value } j \text{ such that } x_j = y\},$$

where the words $x_1, x_2, y$ are binary.

**Theorem 4.20.** *A quantum finite 2-tape automaton exists which recognizes the language $L_4$ with a probability $\frac{4}{7}$.*

*Proof.* The automaton has two accepting $q_{a1}, q_{a2}$ and three rejecting states $q_{r1}, q_{r2}, q_{r3}$ and starts the following actions by reading the pair $(\#, \#)$ with the following amplitudes:

a) with an amplitude $\sqrt{\frac{2}{7}}$ compares $x_1$ to $y$,

b) with an amplitude $-\sqrt{\frac{2}{7}}$ compares $x_2$ to $y$,

c) with an amplitude $\sqrt{\frac{3}{7}}$ immediately goes to the state $q_{a1}$.

Actions a) and b) use different non-halting states to process the word pair. All these actions the automaton processes simultaneously. In actions a) and b), if **no** (not equal), it goes accordingly to the states $q_{r1}$ or $q_{r2}$, if **yes**, then reaches correspondent non-halting states $q_\alpha$ or $q_\beta$, while the symbol pair on the tapes is $(\$, \$)$. The transition for $(\$, \$)$ and states $q_\alpha, q_\beta, q_{a2}, q_{r3}$ is as follows:

| | $q_\alpha$ | $q_\beta$ |
|---|---|---|
| $q_{a2}$ | $\frac{1}{\sqrt{2}}$ | $\frac{1}{\sqrt{2}}$ |
| $q_{r3}$ | $\frac{1}{\sqrt{2}}$ | $-\frac{1}{\sqrt{2}}$ |

If all the words are equal, it is possible to ensure that it takes the same time to reach the end-markers on both tapes, therefore the automaton reaches the superposition $\sqrt{\frac{2}{7}}|q_\alpha, s, t\rangle - \sqrt{\frac{2}{7}}|q_\beta, s, t\rangle$, where $s$ and $t$ specify the place of \$ on each tape, and the input is accepted with probability $\frac{3}{7}$. (Since the amplitudes of the actions a) and b) equal to 0.) If one of the words $x_i$ equals $y$, then the input is accepted with probability $\frac{4}{7}$. If none of the words $x_i$ equals $y$, then the input is accepted with probability $\frac{3}{7}$. □

# Chapter 5

# Quantum Pushdown Automata

## 5.1 Definition of QPA

The purpose of this chapter is to introduce a quantum counterpart of pushdown automata, the next most important model after finite automata and Turing machines. The first definition of quantum pushdown automata was suggested by [MC 97], but here the authors actually deal with the so-called generalized quantum pushdown automata, which evolution does not have to be unitary. However a basic postulate of quantum mechanics imposes a strong constraint on any quantum machine model: it has to be unitary, otherwise it is questionable whether we can speak about *quantum* machine. That's why it was considered necessary to re-introduce quantum pushdown automata by giving a definition which would conform unitarity requirement. Such definition would enable us to study the properties of quantum pushdown automata.

**Definition 5.1.** *A quantum pushdown automaton (QPA)* $A = (Q, \Sigma, T, q_0, Q_a, Q_r, \delta)$ *is specified by a finite set of states* $Q$*, a finite input alphabet* $\Sigma$ *and a pushdown store alphabet* $T$*, an initial state* $q_0 \in Q$*, sets* $Q_a \subset Q$*,* $Q_r \subset Q$ *of accepting and rejecting states, respectively, with* $Q_a \cap Q_r = \emptyset$*, and a transition function*

$$\delta : Q \times \Gamma \times \Delta \times Q \times \{\downarrow, \rightarrow\} \times \Delta^* \longrightarrow \mathbb{C}_{[0,1]},$$

*where* $\Gamma = \Sigma \cup \{\#, \$\}$ *is the input tape alphabet of* $A$ *and* $\#, \$$ *are end-markers not in* $\Sigma$*,* $\Delta = T \cup \{Z_0\}$ *is the working pushdown store alphabet of* $A$ *and* $Z_0 \notin T$ *is the pushdown store base symbol;* $\{\downarrow, \rightarrow\}$ *is the set of directions of input tape head. The automaton must satisfy* conditions of well-formedness, *which will be expressed below. Furthermore, the transition function is restricted to the following requirement:*

*If* $\delta(q, \alpha, \beta, q', d, \omega) \neq 0$, *then*

1. $|\omega| \leq 2$;
2. *if* $|\omega| = 2$, *then* $\omega_1 = \beta$;
3. *if* $\beta = Z_0$, *then* $\omega \in Z_0 T^*$;
4. *if* $\beta \neq Z_0$, *then* $\omega \in T^*$.

Here $\omega_1$ is the first symbol of a word $\omega$. Definition 5.1 utilizes that of classical pushdown automata from [Gu 89].

Let us assume that an automaton is in a state $q$, its input tape head is above a symbol $\alpha$ and the pushdown store head is above a symbol $\beta$. Then the automaton undertakes following actions with an amplitude $\delta(q, \alpha, \beta, q', d, \omega)$:

1. goes into the state $q'$;
2. if $d =$ ' $\rightarrow$ ', moves the input tape head one cell forward;
3. takes out of the pushdown store the symbol $\beta$ (deletes it and moves the pushdown store head one cell backwards);
4. starting with the first empty cell, puts into the pushdown store the string $\omega$, moving the pushdown store head $|\omega|$ cells forward.

**Definition 5.2.** *The* configuration *of a pushdown automaton is a pair* $|c\rangle = |\nu_i q_j \nu_k, \omega_l\rangle$, *where the automaton is in a state* $q_j \in Q$, $\nu_i \nu_k \in \#\Sigma^*\$$ *is a finite word on the input tape,* $\omega_l \in Z_0 T^*$ *is a finite word on the pushdown store tape, the input tape head is above the first symbol of the word* $\nu_k$ *and the pushdown store head is above the last symbol of the word* $\omega_l$.

We shall denote by $C$ the set of all configurations of a pushdown automaton. The set $C$ is countably infinite. Every configuration $|c\rangle$ denotes a basis vector in the Hilbert space $H_A = l_2(C)$. Therefore a global state of $A$ in the space $H_A$ has a form $|\psi\rangle = \sum\limits_{c \in C} \alpha_c |c\rangle$, where $\sum\limits_{c \in C} |\alpha_c|^2 = 1$ and $\alpha_c \in \mathbb{C}$ denotes the amplitude of a configuration $|c\rangle$. If an automaton is in its global state (superposition) $|\psi\rangle$, then its further step is equivalent to the application of a linear operator (evolution) $U_A$ over the space $H_A$.

**Definition 5.3.** *A linear operator* $U_A$ *is defined as follows:*

$$U_A |\psi\rangle = \sum_{c \in C} \alpha_c U_A |c\rangle.$$

*If a configuration* $c = |\nu_i q_j \sigma \nu_k, \omega_l \tau\rangle$, *then*

$$U_A|c\rangle = \sum_{(q,d,\omega)\in Q\times\{\downarrow,\rightarrow\}\times\Delta^*} \delta(q_j,\sigma,\tau,q,d,\omega)|f(|c\rangle,d,q),\omega_l\omega\rangle,$$

*where*

$$f(|\nu_i q_j \sigma \nu_k, \omega_l \tau\rangle, d, q) = \begin{cases} \nu_i q \sigma \nu_k, \textit{ if } d = \text{`}\downarrow\text{'} \\ \nu_i \sigma q \nu_k, \textit{ if } d = \text{`}\rightarrow\text{'}. \end{cases}$$

We can speak about a *quantum* pushdown automaton only if its evolution operator is unitary. However, evolution operator matrix is infinite, so we need some criteria (well-formedness conditions) to verify its unitarity.

**Well-formedness conditions 5.4.**

1. *Local probability condition.*

$$\forall(q_1,\sigma_1,\tau_1) \in Q\times\Gamma\times\Delta$$
$$\sum_{(q,d,\omega)\in Q\times\{\downarrow,\rightarrow\}\times\Delta^*} |\delta(q_1,\sigma_1,\tau_1,q,d,\omega)|^2 = 1. \tag{5.1}$$

2. *Orthogonality of column vectors condition.*

*For all triples* $(q_1,\sigma_1,\tau_1) \neq (q_2,\sigma_1,\tau_2)$ *in* $Q\times\Gamma\times\Delta$

$$\sum_{(q,d,\omega)\in Q\times\{\downarrow,\rightarrow\}\times\Delta^*} \delta^*(q_1,\sigma_1,\tau_1,q,d,\omega)\delta(q_2,\sigma_1,\tau_2,q,d,\omega) = 0. \tag{5.2}$$

3. *Row vectors norm condition.*

$$\forall(q_1,\sigma_1,\sigma_2,\tau_1,\tau_2) \in Q\times\Gamma^2\times\Delta^2$$
$$\sum_{(q,\tau,\omega)\in Q\times\Delta\times\{\epsilon,\tau_2,\tau_1\tau_2\}} |\delta(q,\sigma_1,\tau,q_1,\rightarrow,\omega)|^2 +$$
$$+|\delta(q,\sigma_2,\tau,q_1,\downarrow,\omega)|^2 = 1. \tag{5.3}$$

4. *Separability condition I.*

$$\forall(q_1,\sigma_1,\tau_1),(q_2,\sigma_1,\tau_2) \in Q\times\Gamma\times\Delta,\ \forall\tau_3\in\Delta$$

a) $$\sum_{(q,d,\tau)\in Q\times\{\downarrow,\rightarrow\}\times\Delta} \delta^*(q_1,\sigma_1,\tau_1,q,d,\tau)\delta(q_2,\sigma_1,\tau_2,q,d,\tau_3\tau) +$$
$$+ \sum_{(q,d)\in Q\times\{\downarrow,\rightarrow\}} \delta^*(q_1,\sigma_1,\tau_1,q,d,\epsilon)\delta(q_2,\sigma_1,\tau_2,q,d,\tau_3) = 0; \tag{5.4}$$

b) $$\sum_{(q,d)\in Q\times\{\downarrow,\rightarrow\}} \delta^*(q_1,\sigma_1,\tau_1,q,d,\epsilon)\delta(q_2,\sigma_1,\tau_2,q,d,\tau_2\tau_3) = 0. \tag{5.5}$$

5. *Separability condition II.*

$$\forall (q_1,\sigma_1,\tau_1),(q_2,\sigma_2,\tau_2) \in Q\times\Gamma\times\Delta$$
$$\sum_{(q,\omega)\in Q\times\Delta^*} \delta^*(q_1,\sigma_1,\tau_1,q,\downarrow,\omega)\delta(q_2,\sigma_2,\tau_2,q,\rightarrow,\omega) = 0. \quad (5.6)$$

6. *Separability condition III.*

$$\forall (q_1,\sigma_1,\tau_1),(q_2,\sigma_2,\tau_2) \in Q\times\Gamma\times\Delta,\ \forall \tau_3\in\Delta,$$
$$\forall d_1,d_2\in\{\downarrow,\rightarrow\},\ d_1\neq d_2$$

$$a)\ \sum_{(q,\tau)\in Q\times\Delta} \delta^*(q_1,\sigma_1,\tau_1,q,d_1,\tau)\delta(q_2,\sigma_2,\tau_2,q,d_2,\tau_3\tau) +$$
$$+\sum_{q\in Q}\delta^*(q_1,\sigma_1,\tau_1,q,d_1,\epsilon)\delta(q_2,\sigma_2,\tau_2,q,d_2,\tau_3) = 0; \quad (5.7)$$

$$b)\ \sum_{q\in Q}\delta^*(q_1,\sigma_1,\tau_1,q,d_1,\epsilon)\delta(q_2,\sigma_2,\tau_2,q,d_2,\tau_2\tau_3) = 0. \quad (5.8)$$

**Lemma 5.5.** *The columns system of a QPA evolution matrix is normalized iff the condition (5.1), i.e., local probability condition, is satisfied.*

Condition (5.2) insures that every two columns indexed by configurations of type $|\omega_1 q_1\sigma_1\omega_2,\omega\tau_1\rangle$ and $|\omega_1 q_2\sigma_1\omega_2,\omega\tau_2\rangle$ are orthogonal. Here $(q_1,\tau_1)\neq(q_2,\tau_2)$.

Condition (5.4) insures that every two columns indexed by configurations of type $|\omega_1 q_2\sigma_1\omega_2,\omega\tau_2\rangle$ and $|\omega_1 q_1\sigma_1\omega_2,\omega\tau_3\tau_1\rangle$ are orthogonal.

Condition (5.5) insures that every two columns indexed by configurations of type $|\omega_1 q_2\sigma_1\omega_2,\omega\tau_2\rangle$ and $|\omega_1 q_1\sigma_1\omega_2,\omega\tau_2\tau_3\tau_1\rangle$ are orthogonal.

Condition (5.6) insures that every two columns indexed by configurations of type $|\omega_1 q_2\sigma_2\sigma_1\omega_2,\omega\tau_2\rangle$ and $|\omega_1\sigma_2 q_1\sigma_1\omega_2,\omega\tau_1\rangle$ are orthogonal.

Condition (5.7) insures that every two columns indexed by configurations of type $|\omega_1 q_2\sigma_2\sigma_1\omega_2,\omega\tau_2\rangle$ and $|\omega_1\sigma_2 q_1\sigma_1\omega_2,\omega\tau_3\tau_1\rangle$, or indexed by configurations of type $|\omega_1\sigma_1 q_2\sigma_2\omega_2,\omega\tau_2\rangle$ and $|\omega_1 q_1\sigma_1\sigma_2\omega_2,\omega\tau_3\tau_1\rangle$ are orthogonal.

Condition (5.8) insures that every two columns indexed by configurations of type $|\omega_1 q_2\sigma_2\sigma_1\omega_2,\omega\tau_2\rangle$ and $|\omega_1\sigma_2 q_1\sigma_1\omega_2,\omega\tau_2\tau_3\tau_1\rangle$, or indexed by configurations of type $|\omega_1\sigma_1 q_2\sigma_2\omega_2,\omega\tau_2\rangle$ and $|\omega_1 q_1\sigma_1\sigma_2\omega_2,\omega\tau_2\tau_3\tau_1\rangle$ are orthogonal.

**Lemma 5.6.** *The columns system of a QPA evolution matrix is orthogonal iff the conditions (5.2,5.4,5.5,5.6,5.7,5.8), i.e., orthogonality of column vectors and separability conditions, are satisfied.*

**Lemma 5.7.** *The rows system of a QPA evolution matrix is normalized iff the condition (5.3), i.e., row vectors norm condition, is satisfied.*

**Theorem 5.8.** *Well-formedness conditions 5.4 are satisfied iff the evolution operator $U_A$ is unitary.*

*Proof.* Lemmas 5.5, 5.6, 5.7 imply that Well-formedness conditions 5.4 are satisfied iff the columns of the evolution matrix are orthonormal and rows are normalized. In compliance with Lemma 2.9, columns are orthonormal and rows are normalized iff the matrix is unitary. □

**Remark 5.9.** Well-formedness conditions 5.4 contain the requirement that rows system has to be normalized, which is not necessary in the case of quantum Turing machine [BV 97]. Here is taken into account the fact that the evolution of QPA can violate the unitarity requirement if the row vectors norm condition is omitted.

**Example 5.10.** A QPA, whose evolution matrix columns are orthonormal, however the evolution is not unitary.

$$\begin{array}{ll} Q=\{q\},\ \Sigma=\{1\},\ T=\{1\}. & \\ \delta(q,\#,Z_0,q,\rightarrow,Z_01)=1, & \delta(q,\#,1,q,\rightarrow,11)=1, \\ \delta(q,1,Z_0,q,\rightarrow,Z_01)=1, & \delta(q,1,1,q,\rightarrow,11)=1, \\ \delta(q,\$,Z_0,q,\rightarrow,Z_01)=1, & \delta(q,\$,1,q,\rightarrow,11)=1, \end{array}$$

other values of arguments yield $\delta=0$.

By Well-formedness conditions 5.4, the columns of the evolution matrix are orthonormal, but the matrix is not unitary, because the norm of the rows specified by the configurations $|\omega, Z_0\rangle$ is 0.

**Definition 5.11.** *We say that an automaton is a deterministic reversible pushdown automaton (RPA), if it is a QPA with $\delta(q_1,\sigma,\tau,q,d,\omega)\in\{0,1\}$.*

Even in a case of trivial QPA, it is a cumbersome task to check all the conditions of Well-formedness 5.4. It is possible to relax the conditions slightly by introducing a notion of *simplified* QPA.

**Definition 5.12.** *We shall say that a QPA is* simplified, *if there exists a function $D: Q\longrightarrow\{\downarrow,\rightarrow\}$, and $\delta(q_1,\sigma,\tau,q,d,\omega)=0$, if $D(q)\neq d$. Therefore the transition function of a simplified QPA is*

$$\varphi(q_1,\sigma,\tau,q,\omega)=\delta(q_1,\sigma,\tau,q,D(q),\omega).$$

Taking into account Definition 5.12, following well-formedness conditions correspond to simplified QPA:

**Well-formedness conditions 5.13.**

1. *Local probability condition.*

$$\forall(q_1,\sigma_1,\tau_1)\in Q\times\Gamma\times\Delta \quad \sum_{(q,\omega)\in Q\times\Delta^*}|\varphi(q_1,\sigma_1,\tau_1,q,\omega)|^2=1. \tag{5.9}$$

2. *Orthogonality of column vectors condition.*

*For all triples* $(q_1,\sigma_1,\tau_1)\neq(q_2,\sigma_1,\tau_2)$ *in* $Q\times\Gamma\times\Delta$

$$\sum_{(q,\omega)\in Q\times\Delta^*}\varphi^*(q_1,\sigma_1,\tau_1,q,\omega)\varphi(q_2,\sigma_1,\tau_2,q,\omega)=0. \tag{5.10}$$

3. *Row vectors norm condition.*

$$\forall(q_1,\sigma_1,\tau_1,\tau_2)\in Q\times\Gamma\times\Delta^2 \quad \sum_{(q,\tau,\omega)\in Q\times\Delta\times\{\epsilon,\tau_2,\tau_1\tau_2\}}|\varphi(q,\sigma_1,\tau,q_1,\omega)|^2=1. \tag{5.11}$$

4. *Separability condition.*

$\forall(q_1,\sigma_1,\tau_1),(q_2,\sigma_1,\tau_2)\in Q\times\Gamma\times\Delta,\ \forall\tau_3\in\Delta$

$$a)\ \sum_{(q,\tau)\in Q\times\Delta}\varphi^*(q_1,\sigma_1,\tau_1,q,\tau)\varphi(q_2,\sigma_1,\tau_2,q,\tau_3\tau)+ \\ +\sum_{q\in Q}\varphi^*(q_1,\sigma_1,\tau_1,q,\epsilon)\varphi(q_2,\sigma_1,\tau_2,q,\tau_3)=0; \tag{5.12}$$

$$b)\ \sum_{q\in Q}\varphi^*(q_1,\sigma_1,\tau_1,q,\epsilon)\varphi(q_2,\sigma_1,\tau_2,q,\tau_2\tau_3)=0. \tag{5.13}$$

**Theorem 5.14.** *The evolution of a simplified QPA is unitary iff Well-formedness conditions 5.13 are satisfied.*

*Proof.* By Theorem 5.8 and Definition 5.12. □

**Theorem 5.15.** *The local probability condition (5.9) for simplified RPA is satisfied iff exists a function* $f:Q\times\Gamma\times\Delta\longrightarrow Q\times\Delta^*$, *such that* $f(q_1,\sigma,\tau)=(q,\omega)\Leftrightarrow\varphi(q_1,\sigma,\tau,q,\omega)=1$.

By Theorem 5.15, it is possible to use the function $f$ instead of $\varphi$ and regard $f$ as the transition function of a simplified RPA. Note that the local probability condition (5.9) is now satisfied automatically for simplified RPA.

## 5.2 Language Recognition by QPA

Word acceptance for QPA is defined as in Definition 2.33. For a QPA $A = (Q, \Sigma, T, q_0, Q_a, Q_r, \delta)$ we define $C_a = \{|\nu_i q \nu_k, \omega_l\rangle \in C \mid q \in Q_a\}$, $C_r = \{|\nu_i q \nu_k, \omega_l\rangle \in C \mid q \in Q_r\}$, $C_n = C \setminus (C_a \cup C_r)$. $E_a, E_r, E_n$ are subspaces of $H_A$ spanned by $C_a, C_r, C_n$ respectively. We use the observable $\mathcal{O}$ that corresponds to the orthogonal decomposition $H_A = E_a \oplus E_r \oplus E_n$. The outcome of each observation is either "accept" or "reject" or "non-halting".

For an $x \in \Sigma^*$ we consider as an input $\#x\$$, and assume that the computation starts with $A$ being in the configuration $|q_0\#x\$, Z_0\rangle$. Each computation step consists of two parts. At first the linear operator $U_A$ is applied to the current global state and then the resulting superposition is observed using the observable $\mathcal{O}$ as defined above. If the global state before the observation is $\sum\limits_{c \in C} \alpha_c |c\rangle$, then the probability that the resulting superposition is projected into the subspace $E_i$, $i \in \{a, r, n\}$, is $\sum\limits_{c \in C_i} |\alpha_c|^2$. The computation continues until the result of an observation is "accept" or "reject".

We define language recognition as in Definition 2.37.

**Theorem 5.16.** *Every regular language is recognizable by some QPA.*

*Proof.* It is sufficient to prove that any deterministic finite automaton (DFA) can be simulated by simplified RPA. Let us consider a DFA with $n$ states $A_{DFA} = (Q_{DFA}, \Sigma, q_0, Q_F, \delta)$, where $\delta : Q_{DFA} \times \Sigma \longrightarrow Q_{DFA}$.

To simulate $A_{DFA}$ we shall construct a RPA $A_{RPA} = (Q, \Sigma, T, q_0, Q_a, Q_r, \varphi)$ with the number of states $2n$.

The set of states is $Q = Q_{DFA} \cup Q'_{DFA}$, where $Q_{DFA} \cap Q'_{DFA} = \emptyset$ and $Q'_{DFA}$ are the newly introduced states, which are linked to $Q_{DFA}$ by a one-to-one relation $\{(q_i, q'_i) \in Q_{DFA} \times Q'_{DFA}\}$. Thus $Q_F$ has one-to-one relation to $Q'_F \subset Q'_{DFA}$.

The pushdown store alphabet is $T = \mathrm{ind}(Q_{DFA})$, where $\forall i \ \mathrm{ind}(q_i) = i$; the set of accepting states is $Q_a = Q'_F$ and the set of rejecting states is $Q_r = Q'_{DFA} \setminus Q'_F$. As for the function $D$, $D(Q_{DFA}) = \{\rightarrow\}$ and $D(Q'_{DFA}) = \{\downarrow\}$.

We shall define sets $R$ and $\overline{R}$ as follows:

$$R = \{(q'_j, \sigma, i) \in Q'_{DFA} \times \Sigma \times T \mid \delta(q_i, \sigma) = q_j\};$$

$$\overline{R} = \{(q'_j, \sigma, i) \in Q'_{DFA} \times \Sigma \times T \mid \delta(q_i, \sigma) \neq q_j\}.$$

The construction of the transition function $f$ is performed by the following rules:

1. $\forall (q_i, \sigma, \tau) \in Q_{DFA} \times \Sigma \times \Delta \quad f(q_i, \sigma, \tau) = (\delta(q_i, \sigma), \tau i)$;

2. $\forall(q'_j,\sigma,i) \in R \;\; f(q'_j,\sigma,i) = (q'_i,\epsilon)$;
3. $\forall(q'_j,\sigma,i) \in \overline{R} \;\; f(q'_j,\sigma,i) = (q_j,i)$;
4. $\forall(q'_j,\sigma) \in Q'_{DFA} \times \Sigma \;\; f(q'_j,\sigma,Z) = (q_j,Z)$;
5. $\forall(q,\tau) \in Q \times \Delta \;\; f(q,\#,\tau) = (q,\tau)$;
6. $\forall(q_i,\tau) \in Q_{DFA} \times \Delta \;\; f(q_i,\$,\tau) = (q'_i,\tau)$;
7. $\forall(q'_i,\tau) \in Q'_{DFA} \times \Delta \;\; f(q'_i,\$,\tau) = (q_i,\tau)$.

Thus we have defined $f$ for all the possible arguments. Our automaton simulates the DFA. Note that the automaton may reach a state in $Q'_{DFA}$ only by reading the end-marking symbol \$ on the input tape. As soon as $A_{RPA}$ reaches the end-marking symbol \$, it goes to an accepting state, if its current state is in $Q_F$, and goes to a rejecting state otherwise.

The construction is performed in a way so that $A_{RPA}$ satisfies Well-formedness conditions 5.13.

RPA automatically satisfies the local probability condition (5.9).

Let us prove, that the automaton satisfies the orthogonality condition (5.10).

For RPA, the condition (5.10) is equivalent to the requirement that for all triples $(q_1,\sigma_1,\tau_1) \neq (q_2,\sigma_1,\tau_2)$ $f(q_1,\sigma_1,\tau_1) \neq f(q_2,\sigma_1,\tau_2)$.

If $q_1,q_2 \in Q_{DFA}$, $f(q_1,\sigma_1,\tau_1) \neq f(q_2,\sigma_1,\tau_2)$ by rule 1.

Let us consider the case when $(q_1,\sigma_1,\tau_1),(q_2,\sigma_1,\tau_2) \in R$. We shall denote $q_1,q_2$ as $q'_i,q'_j$ respectively. Let us assume from the contrary that $f(q'_i,\sigma_1,\tau_1) = f(q'_j,\sigma_1,\tau_2)$. By rule 2, $(q'_{\tau_1},\epsilon) = (q'_{\tau_2},\epsilon)$. Hence $\tau_1 = \tau_2$. By the definition of $R$, $\delta(q_{\tau_1},\sigma_1) = q_i$ and $\delta(q_{\tau_2},\sigma_1) = q_j$. Since $\tau_1 = \tau_2$, $q_i = q_j$. Therefore $q'_i = q'_j$, i.e., $q_1 = q_2$. We have come to a contradiction with the fact that $(q_1,\sigma_1,\tau_1) \neq (q_2,\sigma_1,\tau_2)$.

If $(q_1,\sigma_1,\tau_1),(q_2,\sigma_1,\tau_2) \in \overline{R}$, $f(q_1,\sigma_1,\tau_1) \neq f(q_2,\sigma_1,\tau_2)$ by rule 3.

If $q_1 \in Q_{DFA}, q_2 \in Q'_{DFA}$ then $f(q_1,\sigma_1,\tau 1) \neq f(q_2,\sigma_1,\tau_2)$ by rules 1, 2, 3.

In case $\tau_1$ or $\tau_2$ is $Z$, or $\sigma_1 \in \{\#,\$\}$, proof is straightforward.

The compliance with row vectors norm condition (5.11) and separability conditions (5.12) and (5.13) is proved in the same way. □

**Example 5.17.** Let us consider a language $L_1 = (0,1)^*1$, for which we know that it is not recognizable by QFA [KW 97].

This language is recognized by the deterministic finite automaton with two states $q_0,q_1$ and the following transitions: $\delta(q_0,0) = q_0$, $\delta(q_0,1) = q_1$, $\delta(q_1,0) = q_0$, $\delta(q_1,1) = q_1$.

By Theorem 5.16 it is possible to transform this automaton to the following RPA:
$Q = \{q_0, q_1, q_0', q_1'\}$, $Q_a = \{q_1'\}$, $Q_r = \{q_0'\}$, $\Sigma = \{0, 1\}$, $T = \{0, 1\}$,
$D(q_0) = \rightarrow$, $D(q_1) = \rightarrow$, $D(q_0') = \downarrow$, $D(q_1') = \downarrow$.
By the construction rules,

$$\forall q \in Q,\ \forall \sigma \in \Sigma,\ \forall \tau \in \Delta;$$

$$\begin{array}{lll}
f(q_0, 0, \tau) = (q_0, \tau 0), & f(q_1, 0, \tau) = (q_0, \tau 1), & f(q_0, 1, \tau) = (q_1, \tau 0), \\
f(q_1, 1, \tau) = (q_1, \tau 1), & f(q_0', 0, 0) = (q_0', \epsilon), & f(q_1', 1, 0) = (q_0', \epsilon), \\
f(q_0', 0, 1) = (q_1', \epsilon), & f(q_1', 1, 1) = (q_1', \epsilon), & f(q_0', 1, 0) = (q_0, 0), \\
f(q_1', 0, 0) = (q_1, 0), & f(q_0', 1, 1) = (q_0, 1), & f(q_1', 0, 1) = (q_1, 1), \\
f(q_0', \sigma, Z) = (q_0, Z), & f(q_1', \sigma, Z) = (q_1, Z), & f(q, \#, \tau) = (q, \tau), \\
f(q_0, \$, \tau) = (q_0', \tau), & f(q_1, \$, \tau) = (q_1', \tau), & f(q_0', \$, \tau) = (q_0, \tau), \\
f(q_1', \$, \tau) = (q_1, \tau) & &
\end{array}$$

□

Let us consider a language which is not regular, namely,

$$L_2 = \{\omega \in (a, b)^* \mid |\omega|_a = |\omega|_b\},$$

where $|\omega|_i$ denotes the number of occurrences of the symbol $i$ in the word $\omega$.

**Lemma 5.18.** *Language $L_2$ is recognizable by a RPA.*

*Proof.* Our RPA has four states $q_0, q_1, q_2, q_3$, where $q_2$ is an accepting state, whereas $q_3$ - rejecting one. Pushdown store alphabet $T$ consists of two symbols $1, 2$. Pushdown store filled with 1's means that the processed part of the word $\omega$ has more occurrences of a's than b's, whereas 2's means that there are more b's than a's. Furthermore, length of the pushdown store word is equal to the difference of number of a's and b's. Empty pushdown store denotes that the number of a's and b's is equal.

Values of the transition function follow:

$$\begin{array}{lll}
\forall q \in Q \; \forall \tau \in \Delta; & & \\
f(q,\#,\tau) = (q,\tau), & f(q_0,a,Z) = (q_0,Z1), & D(q_0) = \rightarrow, \\
f(q_0,b,Z) = (q_0,Z2), & f(q_0,\$,Z) = (q_2,Z1), & D(q_1) = \downarrow, \\
f(q_0,a,1) = (q_0,11), & f(q_0,b,1) = (q_1,\epsilon), & D(q_2) = \downarrow, \\
f(q_0,\$,1) = (q_3,1), & f(q_0,a,2) = (q_1,\epsilon), & D(q_3) = \downarrow, \\
f(q_0,b,2) = (q_0,22), & f(q_0,\$,2) = (q_3,2), & f(q_1,a,Z) = (q_0,Z), \\
f(q_1,b,Z) = (q_0,Z), & f(q_1,\$,\tau) = (q_1,\tau), & f(q_1,a,1) = (q_3,12), \\
f(q_1,b,1) = (q_0,1), & f(q_1,a,2) = (q_0,2), & f(q_1,b,2) = (q_3,21), \\
f(q_2,a,Z) = (q_3,Z2), & f(q_2,b,Z) = (q_3,Z1), & f(q_2,\$,Z) = (q_0,Z), \\
f(q_2,a,1) = (q_2,\epsilon), & f(q_2,b,1) = (q_0,12), & f(q_2,\$,1) = (q_0,1), \\
f(q_2,a,2) = (q_0,21), & f(q_2,b,2) = (q_2,\epsilon), & f(q_2,\$,2) = (q_0,2), \\
\forall \sigma \in \{a,b,\$\} & f(q_3,\sigma,Z) = (q_3,Z), & \\
f(q_3,a,1) = (q_3,1), & f(q_3,b,1) = (q_3,11), & f(q_3,\$,1) = (q_2,1), \\
f(q_3,a,2) = (q_3,22), & f(q_3,b,2) = (q_3,2), & f(q_3,\$,2) = (q_2,2).
\end{array}$$

□

**Lemma 5.19.** *Pumping lemma for context-free languages. Every context free language $L$ has a positive integer constant $m$ with the following property. If $\omega$ is in $L$ and $|\omega| \geq m$, then $\omega$ can be written as $uvxyz$, where $uv^kxy^kz$ is in $L$ for each $k \geq 0$. Moreover, $|vxy| \leq m$ and $|vy| > 0$.*

The pumping lemma is from [Gu 89], p. 123.

Let us consider a language $L_3$ which is not recognizable by any deterministic pushdown automaton:

**Theorem 5.20.** *Language $L_3 = \{\omega \in (a,b,c)^* | \; |\omega|_a = |\omega|_b = |\omega|_c\}$ is recognizable by a QPA with probability $\frac{2}{3}$.*

*Proof.* Sketch of proof. The automaton takes three equiprobable actions, during the first action it compares $|\omega|_a$ to $|\omega|_b$, whereas during the second action $|\omega|_b$ to $|\omega|_c$ is compared. Input word is rejected if the third action is chosen. Acceptance probability totals $\frac{2}{3}$.

By Lemma 5.19, the language $L_3$ is not a context-free language. (Take $\omega = a^m b^m c^m$) Hence it is not recognizable by deterministic pushdown automata. □

**Theorem 5.21.** *Language $L_4 = \{\omega \in (a,b,c)^* | \; |\omega|_a = |\omega|_b \; xor \; |\omega|_a = |\omega|_c\}$ is recognizable by a QPA with probability $\frac{4}{7}$.*

*Proof.* Sketch of proof. The automaton starts the following actions with the following amplitudes:

a) with an amplitude $\sqrt{\frac{2}{7}}$ compares $|\omega|_a$ to $|\omega|_b$.

b) with an amplitude $-\sqrt{\frac{2}{7}}$ compares $|\omega|_a$ to $|\omega|_c$.

c) with an amplitude $\sqrt{\frac{3}{7}}$ accepts the input. If exactly one comparison gives positive answer, input is accepted with probability $\frac{4}{7}$. If both comparisons gives positive answer, amplitudes, which are chosen to be opposite, annihilate and the input is accepted with probability $\frac{3}{7}$. □

Language $L_4$ cannot be recognized by deterministic pushdown automata. (By Lemma 5.19, take $\omega = a^{m+m!}b^m c^{m+m!}$)

An open problem is to find a language, not recognizable by probabilistic pushdown automata as well.

# Chapter 6

# Conclusion

In this thesis, we introduced the notion of probabilistic reversible automata in general and one-way probabilistic reversible finite automata in particular. We argued, that language recognition properties of probabilistic reversible finite automata are similar to the corresponding properties of quantum finite automata. We proved several closure properties of the language class recognized by classical probabilistic reversible finite automata. We gave the necessary condition for a language to be recognized by classical one-way probabilistic reversible finite automata (i.e, lack of Type 1 and Type 2 constructions). It is conjectured that this condition is also the sufficient one. We showed a clear relationship between Type 1 and Type 2 languages: a language is of Type 1 if and only if reverse of the language is of Type 2. We presented a single construction (Type 0 construction) which generalizes exactly Type 1 and Type 2 constructions.

We gave the classification of one-way reversible finite automata. An interesting question is to prove or disprove the conjectures regarding the hierarchy of language classes recognized by automata classified there.

We introduced quantum finite multitape automata and formulated criteria which ensure unitarity of this model (well-formedness conditions). We presented several non-trivial languages recognized both by quantum and probabilistic finite multitape automata. The question regarding the language class recognized by QFMA is still open.

We introduced quantum pushdown automata and formulated criteria which ensure unitarity of such automata. It is interesting that these criteria are not equivalent to those of quantum Turing machines. We proved that quantum pushdown automata can recognize every regular language by using their pushdown store as a recording device. We presented several non-context-free languages recognized by quantum pushdown automata. An remaining question is to explore, whether the definition discussed in this thesis has advantages over real-time (classical) quantum pushdown automata. It is still unknown, whether quantum pushdown automata have advantages over probabilistic pushdown automata. The language class recognized by quantum pushdown automata is still not very much explored, and the relation of this class to the class of context-free languages is an open problem.

# Bibliography

[AG 00] F. Ablayev, A. Gainutdinova. On the Lower Bounds for One-Way Quantum Automata. *MFCS 2000, Lecture Notes in Computer Science*, Vol. 1893, pp. 132-140, 2000.

[AI 99] M. Amano, K. Iwama. Undecidability on Quantum Finite Automata. *31st STOC*, pp. 368-375, 1999.

[AIR 02] M. Amano, K. Iwama, R. Raymond. Exploiting the Difference in Probability Calculation between Quantum and Probabilistic Computations. *UMC 2002, Lecture Notes in Computer Science*, Vol. 2509, pp. 100-114, 2002.
`http://arxiv.org/abs/quant-ph/0204075`

[Am 02] A. Ambainis. *Personal communication.*

[ABFGK 99] A. Ambainis, R. Bonner, R. Freivalds, M. Golovkins, M. Karpinski. Quantum Finite Multitape Automata. *SOFSEM 1999, Lecture Notes in Computer Science*, Vol. 1725, pp. 340-348, 1999.
`http://arxiv.org/abs/quant-ph/9905026`

[ABFK 99] A. Ambainis, R. Bonner, R. Freivalds, A. Ķikusts. Probabilities to Accept Languages by Quantum Finite Automata. *COCOON 1999, Lecture Notes in Computer Science*, Vol. 1627, pp. 174-183, 1999.
`http://arxiv.org/abs/quant-ph/9904066`

[AF 98] A. Ambainis, R. Freivalds. 1-Way Quantum Finite Automata: Strengths, Weaknesses and Generalizations. *Proc. 39th FOCS*, pp. 332-341, 1998.
`http://arxiv.org/abs/quant-ph/9802062`

[AK 01] A. Ambainis, A. Ķikusts. Exact Results for Accepting Probabilities of Quantum Automata. *MFCS 2001, Lecture Notes in Computer Science*, Vol. 2136, pp. 135-147, 2001.
`http://arxiv.org/abs/quant-ph/0109136`

[AKV 00] A. Ambainis, A. Ķikusts, M. Valdats. On the Class of Languages Recognizable by 1-Way Quantum Finite Automata. *STACS 2001, Lecture Notes in Computer Science*, Vol. 2010, pp. 75-86, 2001.
`http://arxiv.org/abs/quant-ph/0009004`

[ANTV 98] A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani. Dense Quantum Coding and a Lower Bound for 1-Way Quantum Automata. *Proc. 31st STOC*, pp. 376-383, 1999.
`http://arxiv.org/abs/quant-ph/9804043`

[AW 99] A. Ambainis, J. Watrous. Two-Way Finite Automata with Quantum and Classical States.
`http://arxiv.org/abs/cs.CC/9911009`

[An 82] D. Angluin. Inference of Reversible Languages. *Journal of the ACM*, Vol. 29(3), pp. 741-765, 1982.

[BV 97] E. Bernstein, U. Vazirani. Quantum Complexity Theory. *SIAM Journal on Computing*, Vol. 26(5), pp. 1411-1473, 1997.

[BB 01] A. Bērziņa, R. Bonner. Ambainis-Freivalds' Algorithm for Measure-Once Automata. *FCT 2001, Lecture Notes in Computer Science*, Vol. 2138, pp. 83-93, 2001.

[BFK 01] R. Bonner, R. Freivalds, M. Kravtsev. Quantum versus Probabilistic One-Way Finite Automata with Counter. *SOFSEM 2001, Lecture Notes in Computer Science*, Vol. 2234, pp. 181-190, 2001.

[BP 99] A. Brodsky, N. Pippenger. Characterizations of 1-Way Quantum Finite Automata.
`http://arxiv.org/abs/quant-ph/9903014`

[De 85] D. Deutsch. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. *Proceedings of the Royal Society (London)*, Vol. A-400, pp. 97-117, 1985.

[Du 01] A. Dubrovsky. Space-Efficient 1.5-Way Quantum Turing Machine. *FCT 2001, Lecture Notes in Computer Science*, Vol. 2138, pp. 380-383, 2001.

[DSa 96] C. Dürr, M. Santha. A Decision Procedure for Unitary Linear Quantum Cellular Automata. *Proc. 37th FOCS*, pp. 38-45, 1996.
`http://arxiv.org/abs/quant-ph/9604007`

[DSt 89] C. Dwork, L. Stockmeyer. On the Power of 2-Way Probabilistic Finite State Automata. *Proc. 30th FOCS*, pp. 480-485, 1989.

[Fe 82] R. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, Vol. 21(6/7), pp. 467-488, 1982.

[Fr 78] R. Freivalds. Recognition of Languages with High Probability on Different Classes of Automata. *Doklady Akad. Nauk SSSR*, Vol. 239(1), pp. 60-62, 1978. (Russian)
Also *Soviet Math. Doklady*, Vol. 19(2), pp. 295-298, 1978.

[Fr 81] R. Freivalds. Probabilistic Two-Way Machines. *MFCS 1981, Lecture Notes in Computer Science*, Vol. 118, pp. 33-45, 1981.

[FW 01] R. Freivalds, A. Winter. Quantum Finite State Transducers. *SOFSEM 2001, Lecture Notes in Computer Science*, Vol. 2234, pp. 233-242, 2001.
`http://arxiv.org/abs/quant-ph/0011052`

[Go 00] M. Golovkins. Quantum Pushdown Automata. *SOFSEM 2000, Lecture Notes in Computer Science*, Vol. 1963, pp. 336-346, 2000.
`http://arxiv.org/abs/quant-ph/0102054`

[GK 02] M. Golovkins, M. Kravtsev. Probabilistic Reversible Automata and Quantum Automata. *COCOON 2002, Lecture Notes in Computer Science*, Vol. 2387, pp. 574-583, 2002.
`http://arxiv.org/abs/cs.CC/0209018`

[GS 97] C. M. Grinstead, J. L. Snell. Introduction to Probability. *American Mathematical Society*, 1997.
`http://www.dartmouth.edu/~chance/teaching_aids/articles.html`

[Gr 99] J. Gruska. Quantum Challenges. *SOFSEM 1999, Lecture Notes in Computer Science*, Vol. 1725, pp. 1-28, 1999.

[Gu 89] E. Gurari. An Introduction to the Theory of Computation. *Computer Science Press*, 1989.

[HW 79] G. H. Hardy, E. M. Wright. An Introduction to the Theory of Numbers. Fifth Edition. *Oxford University Press*, 1979.

[HS 66] J. Hartmanis, R. E. Stearns. Algebraic Structure Theory of Sequential Machines. *Prentice Hall*, 1966.

[KS 76] J. G. Kemeny, J. L. Snell. Finite Markov Chains. *Springer Verlag*, 1976.

[KW 97] A. Kondacs, J. Watrous. On The Power of Quantum Finite State Automata. *Proc. 38th FOCS*, pp. 66-75, 1997.

[K 99] M. Kravtsev. Quantum Finite One-Counter Automata. *SOFSEM 1999, Lecture Notes in Computer Science*, Vol. 1725, pp. 431-440, 1999.
`http://arxiv.org/abs/quant-ph/9905092`

[MO 79] A. Marshall, I. Olkin. Inequalities: Theory of Majorization and Its Applications. *Academic Press*, 1979.

[M 02] G. Midrijānis. The Complexity of Probabilistic vs Quantum Finite Automata. *Quantum Computation and Learning. 3rd International Workshop. Proceedings*, pp. 61-64, Riga, 2002.

[MC 97] C. Moore, J. P. Crutchfield. Quantum Automata and Quantum Grammars. *Theoretical Computer Science*, Vol. 237(1-2), pp. 275-306, 2000.
`http://arxiv.org/abs/quant-ph/9707031`

[NIHK 01] M. Nakanishi, T. Indoh, K. Hamaguchi, T. Kashiwabara. On the Power of Quantum Pushdown Automata with a Classical Stack and 1.5-Way Quantum Finite Automata. *NAIST Technical Report, NAIST-IS-TR2001005*, 2001.
`http://isw3.aist-nara.ac.jp/IS/TechReport2/report/2001005.ps`

[N 99] A. Nayak. Optimal Lower Bounds for Quantum Automata and Random Access Codes. *Proc. 40th FOCS*, pp. 369-377, 1999.
`http://arxiv.org/abs/quant-ph/9904093`

[PC 01] M. Pica Ciamarra. Quantum Reversibility and a New Model of Quantum Automaton. *FCT 2001, Lecture Notes in Computer Science*, Vol. 2138, pp. 376-379, 2001.
`http://arxiv.org/abs/quant-ph/0102104`

[P 92] J. E. Pin. On Reversible Automata. *LATIN 1992, Lecture Notes in Computer Science*, Vol. 583, pp. 401-416, 1992.

[R 63] M. O. Rabin. Probabilistic Automata. *Information and Control*, Vol. 6(3), pp. 230-245, 1963.

[Sc 01] O. Scegulnaja. Quantum Real-Time Turing Machine. *FCT 2001, Lecture Notes in Computer Science*, Vol. 2138, pp. 412-415, 2001.

[Sh 94] P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. *Proc. 35th FOCS*, pp. 124-134, 1994.
`http://arxiv.org/abs/quant-ph/9508027`

[T 68] G. Thierrin. Permutation Automata. *Mathematical Systems Theory*, Vol. 2(1), pp. 83-90, 1968.

[YKI 01] T. Yamasaki, H. Kobayashi, H. Imai. Two-way Quantum One-Counter Automata.
`http://arxiv.org/abs/cs.CC/0110005`

[YKI 02] T. Yamasaki, H. Kobayashi, H. Imai. Quantum versus Deterministic Counter Automata. *COCOON 2002, Lecture Notes in Computer Science*, vol. 2387, pp. 584-594, 2002.

[YKTI 00] T. Yamasaki, H. Kobayashi, Y. Tokunaga, H. Imai. One-Way Probabilistic Reversible and Quantum One-Counter Automata. *COCOON 2000, Lecture Notes in Computer Science*, Vol. 1858, pp. 436-446, 2000.